

A Dodecahedral Model for Alveoli

Alan D. Freed^{a,b,*}, Shahla Zamani^a, John D. Clayton^b, J. N. Reddy^a

^a*Department of Mechanical Engineering, Texas A&M University, College Station,
TX 77843, United States*

^b*Impact Physics Branch, U.S. Army Research Laboratory, Aberdeen Proving Ground, Aberdeen, MD
21005, United States*

Abstract

write

Keywords:

write

*Corresponding author

Email address: `afreed@tamu.edu`, `alan.d.freed.civ@mail.mil` (Alan D. Freed)

Part 1

Introduction

Injuries that occur after blast or non-penetrating ballistic projectile impacts to a soldier wearing Personal Protective Equipment (PPE) are referred to as Behind Armor Blunt Trauma (BABT). The kinetic energy from such an impact is absorbed by the soldier's PPE, and by the bony and soft tissues of the soldier beneath. Standards have been written by which PPE's have been designed to since 1972. Verification is through experiments where, typically, a suit of body armor is placed over a 'body' subjected to a ballistic impact from a projectile fired by a weapon, all in accordance with a standard. Current practice is to use clay (usually Roma Plastilina No. 1 clay) as a surrogate for the human body in these tests. A principle objective of an internal Army Research Laboratory-Weapons and Materials Research Directorate (ARL-WMRD) project, *Modeling Large Deformations and Stress Wave Mechanics in Soft Biological Tissue*, is to develop mathematical material models for the human body that can then be used in finite element analyses to study BABT. This is a 6.1 research project whose hand-off to a 6.2 development team at project's end will facilitate the Army's design of improved PPE by allowing engineers to run in-silico BABT tests to complement actual in-field experiments.

The ARL-WMRD *Modeling Large Deformations and Stress Wave Mechanics in Soft Biological Tissue* project has three primary objectives: *i*) new material models, *ii*) new experiments, and *iii*) new trauma metrics. Lung has been selected as the soft tissue of interest for this study. What are sought are models and metrics whose parameters are physical and unique, and whose numeric implementation will be efficient and stable. Continuum thermodynamic models for lung tissue and a trauma metric are being developed, viz., [1] and this document. The work done under this sub-project, *A Dodecahedral Model for Alveoli*, complements its parent project, *Modeling Large Deformations and Stress Wave Mechanics in Soft Biological Tissue*, with regards to the first and third objectives of this ARL-WMRD program.

BABT occurs at the microscopic level of alveoli, which make up the parenchyma, i.e., the fleshy tissue of lung that comprises about 95% of lung volume. The objective of this work is to develop mechanistic multi-scale models capable of describing the deformation and damage that occur at an alveolar level, caused by a shock wave traveling through the parenchyma, induced through either a blast or a ballistic impact to a soldier's PPE. In-silico experiments done using this microscopic model are to be used to 'inform' our macroscopic model in those areas where actual lung experiments are difficult if not impossible to perform.

1.1. Problem Statement

Pulmonary contusion is one of the most common thoracic soft-tissue injuries caused by blunt trauma, with a mortality rate of 10-25% [2]. Damage to lungs is the main cause of morbidity following high-level blast exposures [3]. Lung laceration is also common and debilitating [4]. Existing constitutive models for lung tissue were developed from limited

static test data, e.g., [5–7]. These models, and others developed since then, omit relevant physics pertinent to blast and ballistic impacts required to assess BABT. They also require cumbersome optimization protocols to fit non-unique parameter sets [8, 9], and/or are not validated against independent data [10]. Better lung models suitable for dynamic analysis are needed so that the Army can design improved PPE to better protect its soldiers.

The primary objective of the ARL-WMRD project *Modeling Large Deformations and Stress Wave Mechanics in Soft Biological Tissue* is to develop such models for deformation and damage assessment. These are continuum models derived from thermodynamics that utilize internal state variables to account for the irreversible aspects of response [1]. Models (both macroscopic and microscopic) are specifically sought whose parameters are physical and whose parameterization is straightforward. Characterization of the parameters in a model requires experimental data. This presents an enormous challenge, one that is being addressed in the ARL-WMRD project through other university collaborators.

Performing experiments for the purpose of model characterization is extremely difficult when it comes to modeling lung. Lung is a structure; parenchyma is a material. Therefore, one would normally choose to test the parenchyma, and from these data extract one’s model parameters but, because of its spongy nature, we are challenged to do so in a physically meaningful way. Consequently, one typically tests whole lungs, or lobes thereof, and from these structural experiments we are tasked to extract material parameters through an inverse analysis. An alternative approach whereby one could, in principle, acquire parameters for the continuum models being developed at ARL-WMRD would be to homogenize a microscopic structural response for the alveoli of the parenchyma. The work presented here addresses this approach in our modeling of deformation and damage in alveolar structures.

The narrative that follows seeks to develop two material models for lung: one for mechanical deformation and the other for trauma/damage. Models are sought whose parameters have physical interpretation. Ideally, they will enhance our understanding of the deformation and damage mechanisms at play during BABT. Specifically, they will describe how alveoli respond to pressure- and/or shear-wave fronts as these waves pass through them. This modeling will be accomplished by constructing a multi-scale model connecting the parenchyma (macro) and alveolar (micro) levels. In-silico experiments could then be done on the alveolar structural model, whose homogenized response could serve as an aid in the characterization of ARL’s continuum models. The ARL-WMRD models will be designed to perform efficiently for their implementation into finite element codes. This will allow for BABT analyses to be done during the design of future PPE with an ultimate goal of saving soldiers’ lives.

The primary purpose of this work is to provide a microscopic model for lung tissue that can be used as an aid in the parameterization of a macroscopic model for lung that will be reasonably accurate yet efficient to run in full torso finite element analyses to study BABT.

1.2. Approach

Figure 1.1 shows micrographs from a rat lung taken at different magnifications [11]. In the lower-resolution image one sees numerous alveoli that became exposed because of the

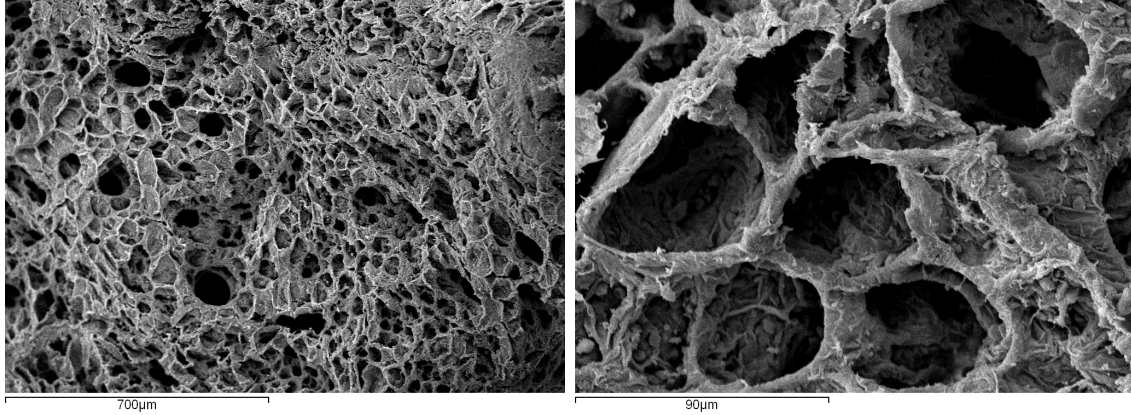


Figure 1.1: SEM photographs from a sectioned rat lung. Left image is at a magnification of 100X. Right image is at a magnification of 750X. They are Figs. 5 & 7 in Freed *et al.* [11]. The alveolar diameter in rat lung is about one quarter the alveolar diameter in human lung.

sectioning process. Also present are several alveolar ducts that connect the individual alveoli with the bronchial tree. In the higher-resolution image we observe the faceted structure of these alveoli, wherein one can see the septal chords and membranes, the latter being traversed by capillaries through which gas exchange occurs. Gas exchange will not modeled here.

The alveolous is modeled here using the geometry of a dodecahedron, a soccer-ball like structure comprised of twelve pentagonal facets bordered by thirty septal cords that are connected at twenty vertices, each vertex linking three neighboring cords of the alveolus with a fourth chord that extends out to neighboring alveoli. BABT occurs, largely, through the tearing of septal cords, which affects the mechanics of breathing, and through the rupturing of capillaries traversing the alveolar facets, causing hemorrhaging thereby leading to lung contusion. A dodecahedral model of alveoli is capable of capturing these trauma events.

Conjecture 1. *A microscopic strain field, measured at the scale of alveoli, is the same as its macroscopic strain field, measured at the scale of parenchyma. Deformation is affine.*

This hypothesis was tested and confirmed in an experimental study done by Butler *et al.* [12] where they used light scattering to study changes in geometry of the septal planes in alveoli from which they concluded: “the microscopic strain field does not differ significantly from the macroscopic field.” We employ this hypothesis by taking the deformation gradient from, say, a Gauss point in a finite element model of lung, and imposing it as a far-field deformation onto our dodecahedral model of an alveolus. From this kinematic input we determine a macroscopic stress response homogenized from the microscopic forces created within this structural model for alveoli.

The authors of a review article on alveolar strain finished by writing:

“In general, computational mechanics approaches to determine function in a healthy or diseased lung have proven to be useful in explaining or measuring observations that are not captured by imaging modalities. However, for these models to fully explain complex physiological mechanical events, appropriate mechanical properties,

boundary conditions, and mechanical loads must be identified. Moreover, validation of such computational models, which is an essential component of any computational mechanics approach, remains to be a challenge in the analysis of soft tissue mechanics.”

Roan & Waters [13, pg. L633]

In this research we set out to develop a constitutive framework for alveolar mechanics, fully cognizant of the aforementioned challenges. Our objectives are different from those of prior studies in alveolar mechanics in that we seek to describe the response/injury of a human lung that has been subjected to a stress wave propagating across the thorax region caused by an impact from either a blunt object or a blast wave. Consequently, some important aspects in the modeling of a breathing lung are thought to be less impactful here, e.g., the effect of surfactant in keeping alveoli from collapsing at the end of expiration.

As a foundation, we adopt the guideline:

“Constitutive equations are phenomenological. They are regarded as empirical by experimenters, and axiomatic by mathematicians. In biomechanics, we often try to derive them on the basis of microstructure . . . in order to gain a better understanding, or to get some guidance to the mathematical form.”

Y.-C. Fung [14, pg. 431]

The approach adopted here is to use the geometry of a dodecahedron as a *microscopic* mechanical model for alveoli, whose far-field response to mechanical stimuli, in accordance with Conjecture 1, will be used to inform the development of a *macroscopic* mechanical model for parenchyma [1], the predominant tissue in lung. This is deemed necessary because of the complex porous structure of parenchyma, as compared with the homogeneous structure of rubbery elastic solids whose theories have historically been employed to model parenchyma [5, 6, 15, 16]. The ARL-WMRD continuum (macroscopic) model for parenchyma [1] will be implemented into finite element codes with an end application being improved and more effective designs for protective body armor.

1.3. Organization

This document is organized in the following manner. Part 2 introduces the dodecahedron as a model for alveoli. Its geometric properties are derived in detail with regards to its three geometric features: 1D septal chords, 2D septal membranes, and 3D volume. Part 3 develops the kinematics required for us to model a deforming dodecahedron, again focusing on the 1D chords, 2D membranes, and the 3D volume. Part 4 derives constitutive models suitable for describing the thermo-mechanical response for the structural constituents of an alveolus: its septal chords, its permeable membranes, and its volume. Part 5 presents numerical methods used for solving first- and second-order ordinary differential equations (ODEs), and for solving areal integrations over a pentagon using a Gaussian quadrature designed for this purpose. Part 6 describes a variational formulation used to create our structural model of an alveolus. And the Appendix provides a reference manual for the software developed for this purpose, which was written in Python vs. 3.7.

Part 2

Dodecahedra: A Model for Alveoli

Typical alveoli are fourteen sided polyhedra with one face normally being open as a mouth to an alveolar duct [17]. To capture the microstructural features of lung, researchers have modeled both alveoli and alveolar ducts, as seen in Fig. 1.1; we only address alveoli here. Three different geometric shapes are typically employed when modeling an alveolus: a dodecahedron introduced by Frankus & Lee [18] in 1974, a rhombic dodecahedron introduced by de Ryk, Thiesse, Namati & McLennan [19] in 2007, and a truncated octahedron, i.e., a tetrakaidecahedron, introduced by Dale, Matthews & Schroter [20] in 1980. The dodecahedron and rhombic dodecahedron are both twelve sided polyhedra with faces being pentagons and rhombuses, respectively. A tetrakaidecahedron is a pair of pyramids stacked bottom to bottom, forming an octahedron, whose six points are then removed. The end result is a fourteen sided polyhedron with six faces that are squares and eight faces that are hexagons, where like shapes have like dimensions.

The tetrakaidecahedron and rhombic dodecahedron are both volume filling. This property is preferred whenever one sets out to construct assemblages of alveoli to build a microstructural model that is to be solved numerically via a finite element method. The purpose of such an exercise is to homogenize the response of an alveolar assembly up to the macroscopic level, i.e., the level of a continuum mass point, a.k.a., the parenchyma [19–25]. Such a finite element model can serve as a representative volume element (RVE) for parenchyma.

The dodecahedron is an isotropic structure, or very nearly so as we shall show, and is nearly volume filling [26]. It becomes the preferred geometry whenever a single alveolus is to be used as the RVE of homogenization, and from which closed form solutions have been derived [11, 26–28]. Here isotropy of the microstructure ensures an isotropic macro response. Parenchyma, as a tissue, is isotropic [16, 29, 30]; whereas, lung, as an organ, is a complex, heterogeneous structure [31, 32]. This distinction has, from time-to-time, gotten lost [33].

For the reasons stated above, a dodecahedron, with vertices labeled according to Fig. 2.1, is the geometric structure selected for use in this study. A dodecahedron contains twelve pentagons. As such, we address the geometry of a pentagon first.

2.1. Properties of a Regular Pentagon

Figure 2.2 presents a regular pentagon drawn in its natural co-ordinate system with coordinates designated as (ξ, η) . Vertices of such a pentagon are placed at

$$\xi = \cos\left(\frac{2(k-1)\pi}{5} + \frac{\pi}{2}\right) \quad \eta = \sin\left(\frac{2(k-1)\pi}{5} + \frac{\pi}{2}\right) \quad k = 1, 2, \dots, 5 \quad (2.1)$$

wherein k denotes the vertex number, as assigned in Fig. 2.2. These vertices inscribe a pentagon within the unit circle.

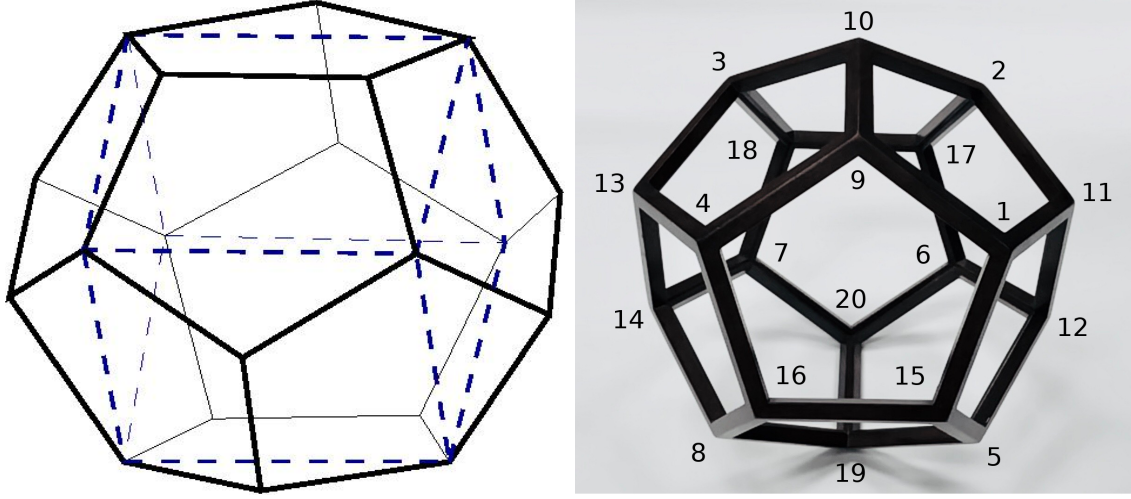


Figure 2.1: Geometric representations for a dodecahedron. Left image: A cube is contained within a dodecahedron, with one of its five possible orientations being displayed. Atop each face of the cube resides four pentagonal sub-areas that form the shape of a hipped roof line. Right image: Vertices 1 through 8 are located at the corners of such a cube. The centroid for the cube is also the centroid for the dodecahedron. Vertices 9 through 20 are corners of the hipped roof lines residing above each face of the cube.

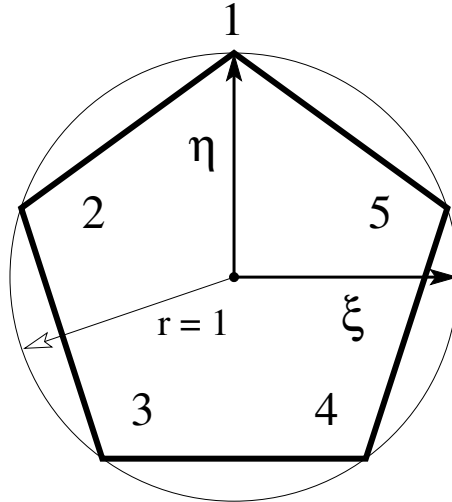


Figure 2.2: A regular pentagon inscribed within the unit circle establishes its natural coordinate system with co-ordinates (ξ, η) described in Eqn. (2.1), and whose origin is located at its centroid. Vertices are numbered counterclockwise with the uppermost vertex being labeled 1.

Lengths of the five chords in a regular pentagon, when measured in its natural co-ordinate system, are all

$$L^p = 2 \cos(\omega) \approx 1.176 \quad (2.2)$$

while the area of this pentagon is

$$A^p = \frac{5}{4} \tan(\omega) L^{p^2} = 5 \sin(\omega) \cos(\omega) \approx 2.378 \quad (2.3)$$

where the area of an unit circle is $\pi r^2 \approx 3.142$ because $r = 1$. The inside angles of a regular pentagon all measure $2\omega = 108^\circ$. All approximations are truncated at four significant figures.

2.1.1. Properties of a Regular Dodecahedron

Like the pentagon considered above, which inscribes the unit circle, here we consider a dodecahedron that inscribes the unit sphere. Let this geometry be described in its natural co-ordinate system with co-ordinates (ξ, η, ζ) whose origin is located at its centroid, the center of the sphere. The twenty vertices of this dodecahedron, all of which touch the unit sphere, are placed at

ξ	η	ζ
$\pm 1/\sqrt{3}$	$\pm 1/\sqrt{3}$	$\pm 1/\sqrt{3}$
$\pm \phi/\sqrt{3}$	$\pm 1/\sqrt{3}\phi$	0
0	$\pm \phi/\sqrt{3}$	$\pm 1/\sqrt{3}\phi$
$\pm 1/\sqrt{3}\phi$	0	$\pm \phi/\sqrt{3}$

(2.4)

where $\phi = (1 + \sqrt{5})/2 \approx 1.618$ is also known as the golden ratio.

Lengths of the thirty chords in a regular dodecahedron, when measured in its natural co-ordinate system, are all

$$L^d = \frac{2}{\sqrt{3}\phi} \approx 0.7136 \quad (2.5)$$

while the volume of such a dodecahedron is

$$V^d = \frac{40}{3\sqrt{3}\phi^3} \tan^2(\omega) \sin(\omega) \approx 2.785 \quad (2.6)$$

where volume of the unit sphere is $\frac{4}{3}\pi r^3 \approx 4.189$ because $r = 1$.

The scale factor to map between the natural co-ordinates of a pentagon, defined in Eq. (2.1), with those of a dodecahedron, defined in Eq. (2.4), is

$$\frac{L^p}{R^p} = \frac{L^d}{R_d^p} \quad \text{or} \quad R_d^p = \frac{R^p L^d}{L^p} = \frac{L^d}{L^p} = \frac{1}{\sqrt{3}\phi \cos(\omega)} \approx 0.6071 \quad (2.7)$$

because $R^p = 1$, with scale factor R_d^p being the radius that inscribes a pentagon on the surface of a dodecahedron that itself inscribes the unit sphere.

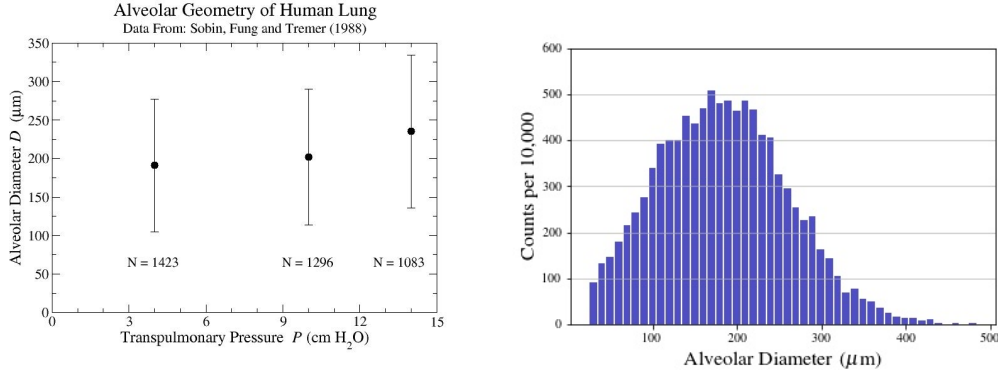


Figure 2.3: Mean and standard deviations for alveolar diameter in human lung illustrated in the left graphic, as reported by Sobin *et al.* [34], with a typical histogram for these statistics illustrated in the right graphic. Note that this distribution has been truncated at an alveolar diameter of 24 μm .

2.1.2. Dimensions of Human Alveoli

Septal chord length $L(D)$, expressed as a function of alveolar diameter D , can be estimated by considering the areal projection of a dodecahedron onto a plane that contains one of its pentagonal faces, which leads to

$$L = \frac{D}{\tan(\omega)(1 + \cos(\alpha))} \approx \frac{D}{2.685}, \quad (2.8)$$

where $\alpha = \pi/10 = 18^\circ$. (There are twenty, equal, pie-shaped wedges that comprise this projected area.) This is an average of the shortest and longest distances across this plane of projection. Alveolar diameter D is a property that can be measured during an histological study of parenchyma.

To dimension the alveoli of human lung, Sobin, Fung & Tremer [34] measured the mean diameter across an individual alveolus, viz., D of Eq. (2.8), sectioned from human lungs that were fixed at three different pressures. Samples were taken from nine lungs extracted postmortem from individuals between 16 to 89 years of age.¹ At a transpulmonary pressure of 4 cm H_2O , the mean alveolar diameter was $D = 191 \pm 86 \mu m$ determined from a sampling size of 1423; at a pressure of 10 cm H_2O , $D = 202 \pm 88 \mu m$ determined from a sampling size of 1296; and at a pressure of 14 cm H_2O , $D = 235 \pm 99 \mu m$ determined from a sampling size of 1083. These data are plotted in Fig. 2.3. All reported and drawn error bounds pertain to plus/minus one standard deviation in error. The distribution was found to be normal.

2.1.3. Geometries of an Irregular Dodecahedron

Formulae (2.3 & 2.6) only apply for regular pentagons and dodecahedra evaluated in their respective natural co-ordinate systems. For irregular dodecahedra, the areas of its irregular

¹Sobin *et al.* [34] documented an age effect in these data that has been averaged over here, i.e., ignored.

pentagons are calculated via²

$$A = \frac{1}{2} \sum_{i=1}^5 (x_i y_{i+1} - x_{i+1} y_i) \quad (2.9)$$

where $x_6 \Leftarrow x_1$ and $y_6 \Leftarrow y_1$. In order for the predicted area to be positive when using this formula, it is necessary that the vertices (x_i, y_i) index counterclockwise, as drawn in Fig. 2.2. The centroid of this pentagon has co-ordinates²

$$c_x = \frac{1}{6A} \sum_{i=1}^5 (x_i + x_{i+1})(x_i y_{i+1} - x_{i+1} y_i) \quad (2.10a)$$

$$c_y = \frac{1}{6A} \sum_{i=1}^5 (y_i + y_{i+1})(x_i y_{i+1} - x_{i+1} y_i) \quad (2.10b)$$

wherein the vertex co-ordinates x_i and y_i are quantified in a 2D pentagonal frame of reference, e.g., as established later in Fig. 2.5.

To compute the volume of an irregular dodecahedron, use the formula³

$$288V_{tet}^2 = \begin{vmatrix} 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & \ell_{12}^2 & \ell_{13}^2 & \ell_{14}^2 \\ 1 & \ell_{21}^2 & 0 & \ell_{23}^2 & \ell_{24}^2 \\ 1 & \ell_{31}^2 & \ell_{32}^2 & 0 & \ell_{34}^2 \\ 1 & \ell_{41}^2 & \ell_{42}^2 & \ell_{43}^2 & 0 \end{vmatrix} \quad (2.11)$$

to calculate each of the 60 individual tetrahedral volumes that collectively fill the volume of an irregular dodecahedron. Here ℓ_{ij} is the length of that tetrahedral edge with vertices i and j ; $i, j = 1, 2, 3, 4$; $i \neq j$; with $\ell_{ij} = \ell_{ji}$.

2.1.4. Indexing Scheme for Dodecahedra

In order to implement the dodecahedron as a geometric model for an alveolar sac, as suggested by the images in Fig. 1.1, it first becomes necessary to introduce a labeling strategy. Such a scheme is arbitrary, but once chosen it enables an analysis to be put forward. The labeling scheme adopted in this work is illustrated in the right image of Fig. 2.1.

The co-ordinates positioning the twenty vertices of a regular dodecahedron in its natural frame of reference are presented in Table 2.1. According to the labeling scheme of Fig. 2.1, the thirty chords of a dodecahedron are given vertex assignments according to Table 2.2, while its twelve pentagons are given vertex assignments according to Table 2.3.

The sixty tetrahedra that fill the volume of the dodecahedron contain vertices according to the following strategy. Beginning with pentagon 1 and sequencing to pentagon 12, two

²Bourke, P., "Polygons, Meshes." <http://paulbourke.net/geometry>.

³Colins, K. D., "Cayley-Menger Determinant." From MathWorld—A Wolfram Web Resource, created by Eric W. Weisstein. <http://mathworld.wolfram.com/CayleyMengerDeterminant.html>.

Vertex	ξ	η	ζ	Vertex	ξ	η	ζ
1	$1/\sqrt{3}$	$1/\sqrt{3}$	$1/\sqrt{3}$	11	$\phi/\sqrt{3}$	$1/\sqrt{3}\phi$	0
2	$1/\sqrt{3}$	$1/\sqrt{3}$	$-1/\sqrt{3}$	12	$\phi/\sqrt{3}$	$-1/\sqrt{3}\phi$	0
3	$-1/\sqrt{3}$	$1/\sqrt{3}$	$-1/\sqrt{3}$	13	$-\phi/\sqrt{3}$	$1/\sqrt{3}\phi$	0
4	$-1/\sqrt{3}$	$1/\sqrt{3}$	$1/\sqrt{3}$	14	$-\phi/\sqrt{3}$	$-1/\sqrt{3}\phi$	0
5	$1/\sqrt{3}$	$-1/\sqrt{3}$	$1/\sqrt{3}$	15	$1/\sqrt{3}\phi$	0	$\phi/\sqrt{3}$
6	$1/\sqrt{3}$	$-1/\sqrt{3}$	$-1/\sqrt{3}$	16	$-1/\sqrt{3}\phi$	0	$\phi/\sqrt{3}$
7	$-1/\sqrt{3}$	$-1/\sqrt{3}$	$-1/\sqrt{3}$	17	$1/\sqrt{3}\phi$	0	$-\phi/\sqrt{3}$
8	$-1/\sqrt{3}$	$-1/\sqrt{3}$	$1/\sqrt{3}$	18	$-1/\sqrt{3}\phi$	0	$-\phi/\sqrt{3}$
9	0	$\phi/\sqrt{3}$	$1/\sqrt{3}\phi$	19	0	$-\phi/\sqrt{3}$	$1/\sqrt{3}\phi$
10	0	$\phi/\sqrt{3}$	$-1/\sqrt{3}\phi$	20	0	$-\phi/\sqrt{3}$	$-1/\sqrt{3}\phi$

Table 2.1: Natural co-ordinates for the vertices of a regular dodecahedron, as labeled in Fig. 2.1 according to Eq. (2.4).

Chord	Vertices	Chord	Vertices	Chord	Vertices
1	9, 10	11	17, 18	21	7, 18
2	1, 9	12	3, 18	22	7, 14
3	2, 10	13	4, 16	23	13, 14
4	3, 10	14	15, 16	24	8, 14
5	4, 9	15	1, 15	25	8, 16
6	1, 11	16	5, 15	26	5, 19
7	2, 11	17	5, 12	27	6, 20
8	3, 13	18	11, 12	28	7, 20
9	4, 13	19	6, 12	29	8, 19
10	2, 17	20	6, 17	30	19, 20

Table 2.2: Vertices that locate the endpoints of septal chords in a dodecahedron, as labeled in Fig. 2.1.

of the four vertices come from a side of the pentagon in question with the remaining two vertices being the centroid for that pentagon and the centroid of the, i.e., at the coordinate origin. From Table 2.3, tetrahedron 1 contains vertices 11 and 2 of pentagon 1, tetrahedron 2 contains vertices 2 and 10, tetrahedron 3 contains vertices 10 and 9, tetrahedron 4 contains vertices 9 and 1, tetrahedron 5 contains vertices 1 and 11, tetrahedron 6 contains vertices 10 and 2 from pentagon 2, etc.

2.1.5. Co-ordinate Systems for Chordal Fibers and Pentagonal Membranes

The dodecahedron used to model an alveolus is considered to be regular in its ‘natural’ configuration, with the capability of being irregular in its reference configuration, and certainly becoming irregular when deformed. The co-ordinate frame of this natural state is taken to have its origin positioned at the centroid of this regular dodecahedron, i.e., at the centroid of an enclosed cube (cf. Fig. 2.1) or, equivalently, at the origin of the unit sphere

Pentagon	Vertices	Chords
1	11, 2, 10, 9, 1	6, 7, 3, 1, 2
2	10, 2, 17, 18, 3	4, 3, 10, 11, 12
3	13, 4, 9, 10, 3	8, 9, 5, 1, 4
4	9, 4, 16, 15, 1	2, 5, 13, 14, 15
5	15, 5, 12, 11, 1	15, 16, 17, 18, 6
6	17, 2, 11, 12, 6	20, 10, 7, 18, 19
7	18, 7, 14, 13, 3	12, 21, 22, 23, 8
8	16, 4, 13, 14, 8	25, 13, 9, 23, 24
9	12, 5, 19, 20, 6	19, 17, 26, 30, 27
10	14, 7, 20, 19, 8	24, 22, 28, 30, 29
11	20, 7, 18, 17, 6	27, 28, 21, 11, 20
12	19, 5, 15, 16, 8	29, 26, 16, 14, 25

Table 2.3: Vertices that locate the corners of regular pentagonal surfaces in a regular dodecahedron, and the chords that connect them. They are indexed counterclockwise when viewed looking from the outside in, and labeled according to Fig. 2.1. The apex for each pentagon resides at the peak of the hipped roof-line for that pentagon. This turns out to be important.

they inscribe, as presented in Table 2.1. We denote the base vectors associated with this frame of reference as $(\vec{i}, \vec{j}, \vec{k})$. There are two other co-ordinate systems with relevance to our analysis: those for the chordal fibers, and those for the pentagonal membranes.

The local co-ordinate system of a chordal fiber is presented in Fig. 2.4, while the local co-ordinate system of a pentagonal membrane is presented in Fig. 2.5. The software described in Appendix E and Appendix F utilize these co-ordinate systems, whose origins are located at their centroids, along with their rotation and spin matrices, which are measured relative to the co-ordinate frame of the dodecahedron, i.e., relative to $(\vec{i}, \vec{j}, \vec{k})$.

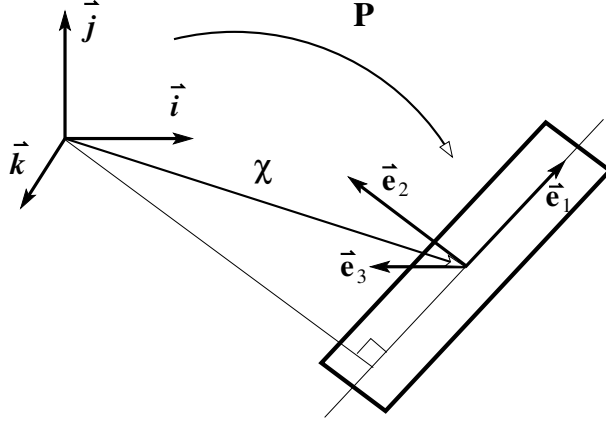


Figure 2.4: The co-ordinate system of a chord ($\vec{e}_1, \vec{e}_2, \vec{e}_3$) relative to the co-ordinate system of its dodecahedron ($\vec{i}, \vec{j}, \vec{k}$) with origins located at their respective centroids that are offset by a translation χ . These describe a mapping $[\{\vec{e}_1\}\{\vec{e}_2\}\{\vec{e}_3\}] = [\{\vec{i}\}\{\vec{j}\}\{\vec{k}\}]\mathbf{P}$ where \mathbf{P} is an orthogonal rotation. The tangent base vector \vec{e}_1 aligns with the axis of this chord. The normal base vector \vec{e}_2 is coaxial with a line segment drawn from the origin out to the chordal axis such that $\vec{e}_1 \cdot \vec{e}_2 = 0$. While the binormal base vector is given by the cross product $\vec{e}_3 = \vec{e}_1 \times \vec{e}_2$.

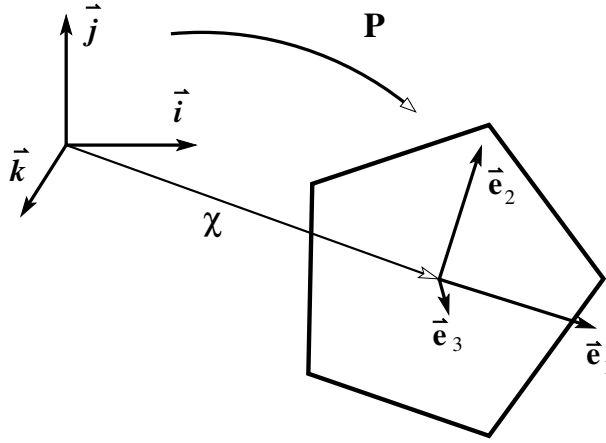


Figure 2.5: The co-ordinate system of a pentagon ($\vec{e}_1, \vec{e}_2, \vec{e}_3$) relative to the co-ordinate system of its dodecahedron ($\vec{i}, \vec{j}, \vec{k}$) with origins located at their respective centroids that are offset by a translation χ . These describe a mapping $[\{\vec{e}_1\}\{\vec{e}_2\}\{\vec{e}_3\}] = [\{\vec{i}\}\{\vec{j}\}\{\vec{k}\}]\mathbf{P}$ where \mathbf{P} is an orthogonal rotation. Base vector \vec{e}_1 is coaxial to a line segment that connects two vertices which locate a pair of shoulders in a pentagon, viz., vertices 2 and 5 in Fig. 2.2. Base vector \vec{e}_2 is coaxial with a line segment drawn from the head of this pentagon, i.e., vertex 1 in Fig. 2.2, down to its base such that $\vec{e}_1 \cdot \vec{e}_2 = 0$. Base vector $\vec{e}_3 = \vec{e}_1 \times \vec{e}_2$ is the outward normal to this surface.

Part 3

Kinematics

The irregular dodecahedron used here as a model for alveoli describes a 3D structure composed of thirty 1D rods (the septal chords) joined at twenty nodes (the vertices) that collectively encompass twelve 2D membranes (the alveolar septa, which are treated here as irregular pentagons). To be able to describe the overall mechanical response of this 3D dodecahedral structure, it is conjectured to be sufficient to know the individual mechanical responses of its 1D septal chords, its 2D septal membranes, and the 3D void within. Their relevant kinematics are presented here.

3.1. 1D Chords

The stretch of a rod under extension is a ratio of its lengths; specifically, $\lambda := L/L_0$ where L and L_0 are its current and reference lengths, respectively. Consequently, the kinematics of chords are trivial.

3.2. 2D Irregular Pentagons

The kinematics of an irregular pentagon, on the other hand, are not trivial. Shape functions are required from which deformation gradients can be constructed. The stretch present in a pentagon can then be derived. Several possible decompositions of the deformation gradient are possible, i.e., stretch is not unique in 2D (nor in 3D). Here we employ the Laplace stretch [35].

3.2.1. Wachspress' Shape Functions for Interpolating an Irregular Pentagon

In 1975, Wachspress [36, 37] derived a set of shape functions N_i that are capable of interpolating convex polyhedra. His shape functions take on the form of rational polynomials, viz., $N_i = A_i/B$ where A_i and B are polynomials. In contrast, classic isoparametric elements are constructed from polynomial shape functions [38]. In the Wachspress shape functions for a pentagon, the A_i are cubic polynomials, while B is a quadratic polynomial.

Let us consider a convex pentagonal domain Ω defined over \mathbb{R}^2 whose vertices have global co-ordinates of

$$(x_1, y_1), (x_2, y_2), (x_3, y_3), (x_4, y_4), (x_5, y_5)$$

when evaluated in the pentagonal co-ordinate system (\vec{e}_1, \vec{e}_2) of Fig. 2.5, with \vec{e}_3 being the outward normal to the pentagon. Associated with this set of global co-ordinates is a set of local or natural co-ordinates

$$(\xi_1, \eta_1), (\xi_2, \eta_2), (\xi_3, \eta_3), (\xi_4, \eta_4), (\xi_5, \eta_5)$$

that describe a mapping of

$$\begin{aligned} x(\xi, \eta) &= \sum_{i=1}^5 N_i(\xi, \eta) x_i & \text{or} & & \mathbf{x}(\boldsymbol{\xi}) &= \sum_{i=1}^5 N_i(\boldsymbol{\xi}) \mathbf{x}_i \\ y(\xi, \eta) &= \sum_{i=1}^5 N_i(\xi, \eta) y_i \end{aligned} \quad (3.1)$$

which relate natural co-ordinates $\boldsymbol{\xi} \equiv (\xi, \eta)$ to global co-ordinates $\mathbf{x} \equiv (x, y)$, where $\mathbf{x}_i \equiv (x_i, y_i)$ are nodal co-ordinates at the i^{th} vertex, with i indexing counterclockwise around a pentagon according to Fig. 2.2. Displacement $\mathbf{u}(\mathbf{x}) := \mathbf{x} - \mathbf{x}_0$, with reference co-ordinates $\mathbf{x}_0 \equiv (x_0, y_0)$, also obeys this mapping

$$\begin{aligned} u(\xi, \eta) &= \sum_{i=1}^5 N_i(\xi, \eta) u_i & \text{or} & & \mathbf{u}(\boldsymbol{\xi}) &= \sum_{i=1}^5 N_i(\boldsymbol{\xi}) \mathbf{u}_i \\ v(\xi, \eta) &= \sum_{i=1}^5 N_i(\xi, \eta) v_i \end{aligned} \quad (3.2)$$

whose co-ordinates $\mathbf{u}_i \equiv (u_i, v_i)$ designate the nodal displacements.

Shape functions $N_i(\boldsymbol{\xi}) \equiv N_i(\xi, \eta)$ are interpolation functions that place any position P with local co-ordinates $\boldsymbol{\xi} \equiv (\xi, \eta) \in \bar{\Omega}$, where $\bar{\Omega} := \Omega \cup \partial\Omega$, into their global co-ordinates $\mathbf{x} \equiv (x, y)$. The shape functions of Wachspress [36, 37] possess the following properties [39]:

1. Partition of unity: $\sum_{i=1}^5 N_i(\boldsymbol{\xi}) = 1$, $0 \leq N_i(\boldsymbol{\xi}) \leq 1$.
2. Interpolate nodal data: $N_i(\boldsymbol{\xi}_j) = \Xi_{ij}$.
3. Linear completeness: $\sum_{i=1}^5 N_i(\boldsymbol{\xi}) \mathbf{x}_i = \mathbf{x}$.
4. For $\boldsymbol{\xi} \in \Omega$, $N_i(\boldsymbol{\xi})$ is C^∞ , but for $\boldsymbol{\xi} \in \partial\Omega$, $N_i(\boldsymbol{\xi})$ is C^0 , i.e., interpolation is linear along an edge (or chord) connecting two neighboring vertices.

For interpolating a convex, planar, pentagonal shape, the shape functions of Wachspress have polynomials of order three in their numerators, and another polynomial of order two in their denominators; specifically, we write them here as

$$N_{i+1}(\xi, \eta) = \kappa_i A_i(\xi, \eta) / B(\xi, \eta), \quad i = 1, 2, \dots, 5 \quad (3.3a)$$

with scaling factors κ_i , where $N_1 \Leftarrow N_6$, whose numerators and denominator for interpolating a pentagon are evaluated via

$$\begin{aligned} A_i(\xi, \eta) &= \alpha_{0i} + \alpha_{1i}\xi + \alpha_{2i}\eta + \alpha_{3i}\xi^2 + \alpha_{4i}\xi\eta + \alpha_{5i}\eta^2 \\ &\quad + \alpha_{6i}\xi^3 + \alpha_{7i}\xi^2\eta + \alpha_{8i}\xi\eta^2 + \alpha_{9i}\eta^3 \end{aligned} \quad (3.3b)$$

$$B(\xi, \eta) = \beta_0 + \beta_1\xi + \beta_2\eta + \beta_3\xi^2 + \beta_4\xi\eta + \beta_5\eta^2 \quad (3.3c)$$

where coefficients in the numerator, i.e., A_i , differ with index i , while those in the denominator, viz., $B := \sum_{i=1}^5 A_i$, are the same for all five shape functions.

We apply the construction technique of Dasgupta [40] to compute the shape functions of Wachspress for an irregular convex pentagon. Let a chord c_i connect vertex $\xi_{i-1} =$

(ξ_{i-1}, η_{i-1}) with vertex $\xi_i = (\xi_i, \eta_i)$ via a straight line segment satisfying $\ell_i = 0$ with $\ell_i := 1 - a_i\xi - b_i\eta$ wherein

$$a_i = \frac{\eta_i - \eta_{i-1}}{\xi_{i-1}\eta_i - \xi_i\eta_{i-1}} \quad (3.4a)$$

$$b_i = \frac{\xi_{i-1} - \xi_i}{\xi_{i-1}\eta_i - \xi_i\eta_{i-1}} \quad (3.4b)$$

for which Dasgupta derived the following set of constraints

$$\kappa_i = \kappa_{i-1} \left(\frac{a_{i+1}(\xi_{i-1} - \xi_i) + b_{i+1}(\eta_{i-1} - \eta_i)}{a_{i-1}(\xi_i - \xi_{i-1}) + b_{i-1}(\eta_i - \eta_{i-1})} \right) \quad (3.4c)$$

with recursion starting at $\kappa_1 := 1$. Coefficients κ_i enforce property 4 listed above.

With this information in hand, we derived the rational polynomials describing Wachpress' shape functions for a pentagon specified in Eq. (3.3) in terms of parameters a_i , b_i and κ_i . The polynomial coefficients for the A_i in Eq. (3.3b) have values of

$$\alpha_{0i} = 1 \quad (3.5a)$$

$$\alpha_{1i} = -(a_{i+1} + a_{i+2} + a_{i+3}) \quad (3.5b)$$

$$\alpha_{2i} = -(b_{i+1} + b_{i+2} + b_{i+3}) \quad (3.5c)$$

$$\alpha_{3i} = a_{i+1}a_{i+2} + a_{i+2}a_{i+3} + a_{i+3}a_{i+1} \quad (3.5d)$$

$$\alpha_{4i} = a_{i+1}(b_{i+2} + b_{i+3}) + a_{i+2}(b_{i+1} + b_{i+3}) + a_{i+3}(b_{i+1} + b_{i+2}) \quad (3.5e)$$

$$\alpha_{5i} = b_{i+1}b_{i+2} + b_{i+2}b_{i+3} + b_{i+3}b_{i+1} \quad (3.5f)$$

$$\alpha_{6i} = -a_{i+1}a_{i+2}a_{i+3} \quad (3.5g)$$

$$\alpha_{7i} = -(a_{i+1}a_{i+2}b_{i+3} + a_{i+1}b_{i+2}a_{i+3} + b_{i+1}a_{i+2}a_{i+3}) \quad (3.5h)$$

$$\alpha_{8i} = -(a_{i+1}b_{i+2}b_{i+3} + b_{i+1}a_{i+2}b_{i+3} + b_{i+1}b_{i+2}a_{i+3}) \quad (3.5i)$$

$$\alpha_{9i} = -b_{i+1}b_{i+2}b_{i+3} \quad (3.5j)$$

which differ for each shape function via index $i = 1, 2, \dots, 5$, while the polynomial coefficients for B in Eq. (3.3c) have values of

$$\beta_j = \sum_{i=1}^5 \alpha_{ji} \kappa_i, \quad j = 0, 1, \dots, 5 \quad (3.6)$$

which are the same for all five shape functions. Sums over the four cubic terms in Eq. (3.5) all vanish—a byproduct of Wachpress' formulation. In the above formulæ, an index count of $i \equiv 0 \implies i = 5$, while index counts of $i \equiv 6 \implies i = 1$, $i \equiv 7 \implies i = 2$ and $i \equiv 8 \implies i = 3$. Shape function N_1 is illustrated in Fig. 3.1, with like images applying for the other four shape functions.

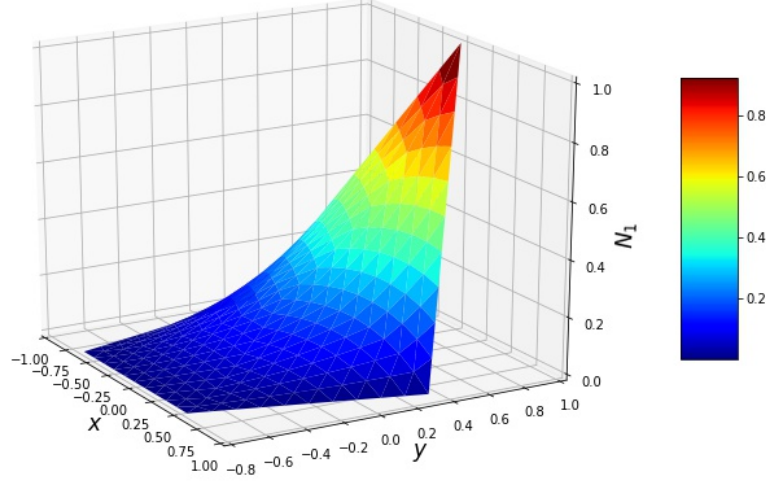


Figure 3.1: Wachspress shape functions for a pentagon, in this case, shape function N_1 .

3.2.2. Spatial Derivatives of Shape Functions

The first derivatives of Wachspress' shape functions for a pentagon are

$$N_{i+1,\xi}(\xi, \eta) = \kappa_i \mathcal{N}_{i,\xi}(\xi, \eta) / B^2(\xi, \eta) \quad (3.7a)$$

$$N_{i+1,\eta}(\xi, \eta) = \kappa_i \mathcal{N}_{i,\eta}(\xi, \eta) / B^2(\xi, \eta) \quad (3.7b)$$

where $N_{i+1,\xi}(\xi, \eta) = \partial N_{i+1}(\xi, \eta) / \partial \xi$ and $N_{i+1,\eta}(\xi, \eta) = \partial N_{i+1}(\xi, \eta) / \partial \eta$ with

$$\mathcal{N}_{i,\xi}(\xi, \eta) = B(\xi, \eta) A_{i,\xi}(\xi, \eta) - B_{,\xi}(\xi, \eta) A_i(\xi, \eta) \quad (3.7c)$$

$$\mathcal{N}_{i,\eta}(\xi, \eta) = B(\xi, \eta) A_{i,\eta}(\xi, \eta) - B_{,\eta}(\xi, \eta) A_i(\xi, \eta) \quad (3.7d)$$

which contain the polynomials

$$A_{i,\xi}(\xi, \eta) = \alpha_{1i} + 2\alpha_{3i}\xi + \alpha_{4i}\eta + 3\alpha_{6i}\xi^2 + 2\alpha_{7i}\xi\eta + \alpha_{8i}\eta^2 \quad (3.7e)$$

$$A_{i,\eta}(\xi, \eta) = \alpha_{2i} + \alpha_{4i}\xi + 2\alpha_{5i}\eta + \alpha_{7i}\xi^2 + 2\alpha_{8i}\xi\eta + 3\alpha_{9i}\eta^2 \quad (3.7f)$$

$$B_{,\xi}(\xi, \eta) = \beta_1 + 2\beta_3\xi + \beta_4\eta \quad (3.7g)$$

$$B_{,\eta}(\xi, \eta) = \beta_2 + \beta_4\xi + 2\beta_5\eta \quad (3.7h)$$

from which the deformation and displacement gradients are constructed.

The second derivatives of these shape functions, which appear in the compatibility equations, are described by

$$N_{i+1,\xi\xi} = \kappa_i \mathfrak{N}_{i,\xi\xi}(\xi, \eta)/B^3(\xi, \eta) \quad (3.8a)$$

$$N_{i+1,\xi\eta} = \kappa_i \mathfrak{N}_{i,\xi\eta}(\xi, \eta)/B^3(\xi, \eta) \quad (3.8b)$$

$$N_{i+1,\eta\xi} = \kappa_i \mathfrak{N}_{i,\eta\xi}(\xi, \eta)/B^3(\xi, \eta) \quad (3.8c)$$

$$N_{i+1,\eta\eta} = \kappa_i \mathfrak{N}_{i,\eta\eta}(\xi, \eta)/B^3(\xi, \eta) \quad (3.8d)$$

where $N_{i+1,\xi\eta}(\xi, \eta) = \partial^2 N_{i+1}(\xi, \eta)/\partial\xi\partial\eta$, etc., and where

$$\mathfrak{N}_{i,\xi\xi}(\xi, \eta) = B(\xi, \eta)\mathcal{N}_{i,\xi\xi}(\xi, \eta) - 2B_{,\xi}(\xi, \eta)\mathcal{N}_{i,\xi}(\xi, \eta) \quad (3.8e)$$

$$\mathfrak{N}_{i,\xi\eta}(\xi, \eta) = B(\xi, \eta)\mathcal{N}_{i,\xi\eta}(\xi, \eta) - 2B_{,\xi}(\xi, \eta)\mathcal{N}_{i,\eta}(\xi, \eta) \quad (3.8f)$$

$$\mathfrak{N}_{i,\eta\xi}(\xi, \eta) = B(\xi, \eta)\mathcal{N}_{i,\eta\xi}(\xi, \eta) - 2B_{,\eta}(\xi, \eta)\mathcal{N}_{i,\xi}(\xi, \eta) \quad (3.8g)$$

$$\mathfrak{N}_{i,\eta\eta}(\xi, \eta) = B(\xi, \eta)\mathcal{N}_{i,\eta\eta}(\xi, \eta) - 2B_{,\eta}(\xi, \eta)\mathcal{N}_{i,\eta}(\xi, \eta) \quad (3.8h)$$

wherein

$$\mathcal{N}_{i,\xi\xi}(\xi, \eta) = B(\xi, \eta)A_{i,\xi\xi}(\xi, \eta) - B_{,\xi\xi}(\xi, \eta)A_i(\xi, \eta) \quad (3.8i)$$

$$\begin{aligned} \mathcal{N}_{i,\xi\eta}(\xi, \eta) &= B(\xi, \eta)A_{i,\xi\eta}(\xi, \eta) + B_{,\xi}(\xi, \eta)A_{i,\eta}(\xi, \eta) \\ &\quad - B_{,\eta}(\xi, \eta)A_{i,\xi}(\xi, \eta) - B_{,\xi\eta}(\xi, \eta)A_i(\xi, \eta) \end{aligned} \quad (3.8j)$$

$$\begin{aligned} \mathcal{N}_{i,\eta\xi}(\xi, \eta) &= B(\xi, \eta)A_{i,\eta\xi}(\xi, \eta) + B_{,\eta}(\xi, \eta)A_{i,\xi}(\xi, \eta) \\ &\quad - B_{,\xi}(\xi, \eta)A_{i,\eta}(\xi, \eta) - B_{,\eta\xi}(\xi, \eta)A_i(\xi, \eta) \end{aligned} \quad (3.8k)$$

$$\mathcal{N}_{i,\eta\eta}(\xi, \eta) = B(\xi, \eta)A_{i,\eta\eta}(\xi, \eta) - B_{,\eta\eta}(\xi, \eta)A_i(\xi, \eta) \quad (3.8l)$$

which contain polynomials

$$A_{i,\xi\xi}(\xi, \eta) = 2\alpha_{3i} + 6\alpha_{6i}\xi + 2\alpha_{7i}\eta \quad (3.8m)$$

$$A_{i,\xi\eta}(\xi, \eta) = \alpha_{4i} + 2\alpha_{7i}\xi + 2\alpha_{8i}\eta \quad (3.8n)$$

$$A_{i,\eta\eta}(\xi, \eta) = 2\alpha_{5i} + 2\alpha_{8i}\xi + 6\alpha_{9i}\eta \quad (3.8o)$$

$$B_{,\xi\xi}(\xi, \eta) = 2\beta_3 \quad (3.8p)$$

$$B_{,\xi\eta}(\xi, \eta) = \beta_4 \quad (3.8q)$$

$$B_{,\eta\eta}(\xi, \eta) = 2\beta_5 \quad (3.8r)$$

with $A_{i,\xi\eta}(\xi, \eta) = A_{i,\eta\xi}(\xi, \eta)$ and $B_{,\xi\eta}(\xi, \eta) = B_{,\eta\xi}(\xi, \eta)$.

3.2.3. Deformation Gradient for an Irregular Pentagon

Derivatives of displacement (u, v) taken with respect to the local co-ordinates (ξ, η) used to describe the shape functions $N_i(\xi, \eta)$ of a pentagon result in a local displacement gradient with components

$$\begin{bmatrix} \partial u/\partial\xi & \partial u/\partial\eta \\ \partial v/\partial\xi & \partial v/\partial\eta \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^5 N_{i,\xi}(\xi, \eta) u_i & \sum_{i=1}^5 N_{i,\eta}(\xi, \eta) u_i \\ \sum_{i=1}^5 N_{i,\xi}(\xi, \eta) v_i & \sum_{i=1}^5 N_{i,\eta}(\xi, \eta) v_i \end{bmatrix} \quad (3.9a)$$

where $u = x - x_0$ and $v = y - y_0$. Gradients of the global co-ordinates (x_0, y_0) evaluated in a reference state taken with respect to local co-ordinates (ξ, η) are described by the matrix equation

$$\begin{bmatrix} \partial x_0 / \partial \xi & \partial x_0 / \partial \eta \\ \partial y_0 / \partial \xi & \partial y_0 / \partial \eta \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^5 N_{i,\xi}(\xi, \eta) x_{0i} & \sum_{i=1}^5 N_{i,\eta}(\xi, \eta) x_{0i} \\ \sum_{i=1}^5 N_{i,\xi}(\xi, \eta) y_{0i} & \sum_{i=1}^5 N_{i,\eta}(\xi, \eta) y_{0i} \end{bmatrix} \quad (3.9b)$$

wherein (x_{0i}, y_{0i}) are the reference global co-ordinates at the i^{th} vertex, while gradients of the global co-ordinates (x, y) evaluated in the current state taken with respect to local co-ordinates (ξ, η) are described by the matrix equation

$$\begin{bmatrix} \partial x / \partial \xi & \partial x / \partial \eta \\ \partial y / \partial \xi & \partial y / \partial \eta \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^5 N_{i,\xi}(\xi, \eta) x_i & \sum_{i=1}^5 N_{i,\eta}(\xi, \eta) x_i \\ \sum_{i=1}^5 N_{i,\xi}(\xi, \eta) y_i & \sum_{i=1}^5 N_{i,\eta}(\xi, \eta) y_i \end{bmatrix} \quad (3.9c)$$

wherein (x_i, y_i) are the current global co-ordinates at the i^{th} vertex.

From the above matrices, one can construct the deformation gradient $\mathbf{F} = \partial \mathbf{x} / \partial \mathbf{x}_0 = \mathbf{I} + \partial \mathbf{u} / \partial \mathbf{x}_0$ for an irregular pentagon via

$$\mathbf{F}(\xi, \eta) = \begin{bmatrix} F_{11}(\xi, \eta) & F_{12}(\xi, \eta) \\ F_{21}(\xi, \eta) & F_{22}(\xi, \eta) \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} + \begin{bmatrix} \partial u / \partial \xi & \partial u / \partial \eta \\ \partial v / \partial \xi & \partial v / \partial \eta \end{bmatrix} \begin{bmatrix} \partial x_0 / \partial \xi & \partial x_0 / \partial \eta \\ \partial y_0 / \partial \xi & \partial y_0 / \partial \eta \end{bmatrix}^{-1} \quad (3.10)$$

and its associated displacement gradient $\mathbf{G} = \partial \mathbf{u} / \partial \mathbf{x}$ as

$$\mathbf{G}(\xi, \eta) = \begin{bmatrix} G_{11}(\xi, \eta) & G_{12}(\xi, \eta) \\ G_{21}(\xi, \eta) & G_{22}(\xi, \eta) \end{bmatrix} = \begin{bmatrix} \partial u / \partial \xi & \partial u / \partial \eta \\ \partial v / \partial \xi & \partial v / \partial \eta \end{bmatrix} \begin{bmatrix} \partial x / \partial \xi & \partial x / \partial \eta \\ \partial y / \partial \xi & \partial y / \partial \eta \end{bmatrix}^{-1} \quad (3.11)$$

which are evaluated in the 1-2 plane of a co-ordinate system $(\vec{\mathbf{e}}_1, \vec{\mathbf{e}}_2, \vec{\mathbf{e}}_3)$ that orients this pentagon, as illustrated in Fig. 2.5. The deformation and displacement gradients are the two, fundamental, kinematic fields most commonly used in the construction of constitutive equations. They are exported as methods from module `shapeFunctions` described in Appendix F.1.

3.2.4. Compatibility Conditions

To ensure that a deformation is compatible, and therefore integrable, it follows that the curl of the deformation gradient is zero [41]. This produces two constraint equations for the case of planar membranes, they being

$$F_{11,2} = F_{12,1} \quad \text{and} \quad F_{22,1} = F_{21,2} \quad (3.12)$$

whose spatial derivatives associate with the $(\vec{\mathbf{e}}_1, \vec{\mathbf{e}}_2)$ co-ordinate frame.

From Eqn. (3.10), it follows that the spatial derivatives of the deformation gradient are

$$\begin{aligned} \mathbf{F}_{,1}(\xi, \eta) &= \frac{\partial}{\partial x_0} \begin{bmatrix} F_{11}(\xi, \eta) & F_{12}(\xi, \eta) \\ F_{21}(\xi, \eta) & F_{22}(\xi, \eta) \end{bmatrix} \\ &= \frac{\partial \xi}{\partial x_0} \left(\frac{\partial}{\partial \xi} \left(\begin{bmatrix} \partial u / \partial \xi & \partial u / \partial \eta \\ \partial v / \partial \xi & \partial v / \partial \eta \end{bmatrix} \begin{bmatrix} \partial x_0 / \partial \xi & \partial x_0 / \partial \eta \\ \partial y_0 / \partial \xi & \partial y_0 / \partial \eta \end{bmatrix}^{-1} - \begin{bmatrix} \partial u / \partial \xi & \partial u / \partial \eta \\ \partial v / \partial \xi & \partial v / \partial \eta \end{bmatrix} \right. \right. \\ &\quad \left. \left. \times \begin{bmatrix} \partial x_0 / \partial \xi & \partial x_0 / \partial \eta \\ \partial y_0 / \partial \xi & \partial y_0 / \partial \eta \end{bmatrix}^{-1} \frac{\partial}{\partial \xi} \left(\begin{bmatrix} \partial x_0 / \partial \xi & \partial x_0 / \partial \eta \\ \partial y_0 / \partial \xi & \partial y_0 / \partial \eta \end{bmatrix} \begin{bmatrix} \partial x_0 / \partial \xi & \partial x_0 / \partial \eta \\ \partial y_0 / \partial \xi & \partial y_0 / \partial \eta \end{bmatrix}^{-1} \right) \right) \right) \end{aligned} \quad (3.13a)$$

and

$$\begin{aligned}
\mathbf{F}_{,2}(\xi, \eta) &= \frac{\partial}{\partial y_0} \begin{bmatrix} F_{11}(\xi, \eta) & F_{12}(\xi, \eta) \\ F_{21}(\xi, \eta) & F_{22}(\xi, \eta) \end{bmatrix} \\
&= \frac{\partial \eta}{\partial y_0} \left(\frac{\partial}{\partial \eta} \left(\begin{bmatrix} \partial u / \partial \xi & \partial u / \partial \eta \\ \partial v / \partial \xi & \partial v / \partial \eta \end{bmatrix} \right) \begin{bmatrix} \partial x_0 / \partial \xi & \partial x_0 / \partial \eta \\ \partial y_0 / \partial \xi & \partial y_0 / \partial \eta \end{bmatrix}^{-1} - \begin{bmatrix} \partial u / \partial \xi & \partial u / \partial \eta \\ \partial v / \partial \xi & \partial v / \partial \eta \end{bmatrix} \right. \\
&\quad \times \left. \begin{bmatrix} \partial x_0 / \partial \xi & \partial x_0 / \partial \eta \\ \partial y_0 / \partial \xi & \partial y_0 / \partial \eta \end{bmatrix}^{-1} \frac{\partial}{\partial \eta} \left(\begin{bmatrix} \partial x_0 / \partial \xi & \partial x_0 / \partial \eta \\ \partial y_0 / \partial \xi & \partial y_0 / \partial \eta \end{bmatrix} \right) \begin{bmatrix} \partial x_0 / \partial \xi & \partial x_0 / \partial \eta \\ \partial y_0 / \partial \xi & \partial y_0 / \partial \eta \end{bmatrix}^{-1} \right) \quad (3.13b)
\end{aligned}$$

wherein

$$\frac{\partial}{\partial \xi} \begin{bmatrix} \partial u / \partial \xi & \partial u / \partial \eta \\ \partial v / \partial \xi & \partial v / \partial \eta \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^5 N_{i,\xi\xi}(\xi, \eta) u_i & \sum_{i=1}^5 N_{i,\xi\eta}(\xi, \eta) u_i \\ \sum_{i=1}^5 N_{i,\xi\xi}(\xi, \eta) v_i & \sum_{i=1}^5 N_{i,\xi\eta}(\xi, \eta) v_i \end{bmatrix} \quad (3.14a)$$

$$\frac{\partial}{\partial \eta} \begin{bmatrix} \partial u / \partial \xi & \partial u / \partial \eta \\ \partial v / \partial \xi & \partial v / \partial \eta \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^5 N_{i,\eta\xi}(\xi, \eta) u_i & \sum_{i=1}^5 N_{i,\eta\eta}(\xi, \eta) u_i \\ \sum_{i=1}^5 N_{i,\eta\xi}(\xi, \eta) v_i & \sum_{i=1}^5 N_{i,\eta\eta}(\xi, \eta) v_i \end{bmatrix} \quad (3.14b)$$

and

$$\frac{\partial}{\partial \xi} \begin{bmatrix} \partial x_0 / \partial \xi & \partial x_0 / \partial \eta \\ \partial y_0 / \partial \xi & \partial y_0 / \partial \eta \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^5 N_{i,\xi\xi}(\xi, \eta) x_{0i} & \sum_{i=1}^5 N_{i,\xi\eta}(\xi, \eta) x_{0i} \\ \sum_{i=1}^5 N_{i,\xi\xi}(\xi, \eta) y_{0i} & \sum_{i=1}^5 N_{i,\xi\eta}(\xi, \eta) y_{0i} \end{bmatrix} \quad (3.14c)$$

$$\frac{\partial}{\partial \eta} \begin{bmatrix} \partial x_0 / \partial \xi & \partial x_0 / \partial \eta \\ \partial y_0 / \partial \xi & \partial y_0 / \partial \eta \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^5 N_{i,\eta\xi}(\xi, \eta) x_{0i} & \sum_{i=1}^5 N_{i,\eta\eta}(\xi, \eta) x_{0i} \\ \sum_{i=1}^5 N_{i,\eta\xi}(\xi, \eta) y_{0i} & \sum_{i=1}^5 N_{i,\eta\eta}(\xi, \eta) y_{0i} \end{bmatrix} \quad (3.14d)$$

with $\partial \xi / \partial x_0$ and $\partial \eta / \partial y_0$ effectively being scaling factors that we take to be described as a ratio of septal chord lengths; specifically, let

$$\frac{\partial \xi}{\partial x_0} \simeq \frac{\partial \eta}{\partial y_0} \approx \frac{L(\xi, \eta)}{L_0(x, y)} = \frac{\cos(\omega)}{\sqrt{A_0/5 \tan(\omega)}} \quad (3.15)$$

where $L(\xi, \eta)$ is the septal length of a pentagonal edge in its natural configuration, as drawn in Fig. 2.2, while $L_0(x, y)$ is the actual, alveolar, septal length with $A_0(x, y)$ being the area of an alveolar septum in its reference state. This formula follows from Eqns. (2.2 & 2.3).

We study compatibility only for the purpose of assessing applicability of the Wachspress shape functions. It is not otherwise needed for our modeling of an alveolus via a dodecahedron.

3.2.5. Gram-Schmidt Decomposition of the Deformation Gradient

To describe the kinematics of a planar membrane, an upper-triangular Gram-Schmidt decomposition of the deformation gradient \mathbf{F} is used in lieu of the symmetric polar decomposition that is commonly adopted [35, 42–45]. McLellan [46, 47] was the first to propose a triangular decomposition of \mathbf{F} , to prove its uniqueness and existence, and to establish many of its physical properties. This idea has been reinvented several times since then, e.g.,

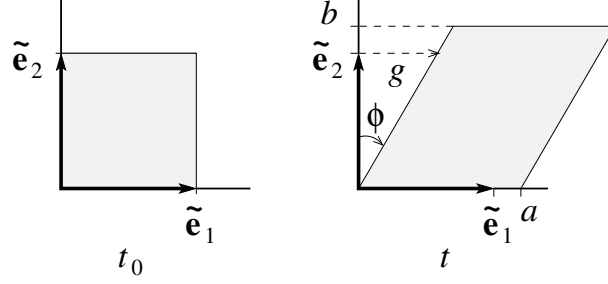


Figure 3.2: Physical attributes of a planar deformation: a and b represent elongations, while $g = \tan \phi$ denotes the extent of shear. They are measured in a physical frame of reference with unit base vectors $(\tilde{\mathbf{e}}_1, \tilde{\mathbf{e}}_2)$ where $\tilde{\mathbf{e}}_1$ embeds in the material.

[42, 48, 49]. A brief history of the **QR** (Gram-Schmidt) decomposition, and its application to kinematics can be found in Ref. [35].

The Gram-Schmidt factorization of a deformation gradient \mathbf{F} is written here as $\mathbf{F} = \mathbf{R}\mathbf{U}$, where the rotation \mathbf{R} is orthogonal, and where the Laplace stretch \mathbf{U} is upper-triangular [35].⁴ This triangular measure of stretch possesses an inherent property in two space: the direction aligned with the rotated 1-axis, denoted as $\tilde{\mathbf{e}}_1$, remains invariant under transformation \mathbf{U} [47], i.e., it is a material vector in a neighborhood surrounding that particle whereat \mathbf{F} is evaluated [50]. This property has ramifications addressed in §3.2.5.2.

3.2.5.1. **QR** Factorization of \mathbf{F}

The 2×2 deformation gradient associated with a planar membrane has a Gram-Schmidt decomposition expressed in terms of four physical attributes. Three of these attributes describe deformation. They are defined as [44]

$$a = \sqrt{F_{11}^2 + F_{21}^2}, \quad b = \frac{F_{11}F_{22} - F_{12}F_{21}}{\sqrt{F_{11}^2 + F_{21}^2}}, \quad g = \frac{F_{11}F_{12} + F_{22}F_{21}}{F_{11}^2 + F_{21}^2} \quad (3.16)$$

thereby populating Laplace stretch \mathbf{U} and its inverse \mathbf{U}^{-1} with components

$$[\mathbf{U}] = \begin{bmatrix} a & ag \\ 0 & b \end{bmatrix} \quad \text{and} \quad [\mathbf{U}]^{-1} = \begin{bmatrix} 1/a & -g/b \\ 0 & 1/b \end{bmatrix} \quad (3.17)$$

where a and b are the principal elongations and g is the extent of in-plane shear, as measured in a co-ordinate frame $(\tilde{\mathbf{e}}_1, \tilde{\mathbf{e}}_2)$ illustrated in Fig. 3.2. It is worth pointing out that the components of Laplace stretch, viz., U_{ij} , are evaluated in the reference co-ordinate system $(\tilde{\mathbf{e}}_1, \tilde{\mathbf{e}}_2)$ of the pentagon, because $\mathbf{F} = F_{ij} \tilde{\mathbf{e}}_i \otimes \tilde{\mathbf{e}}_j$.

⁴The **QR** rotation \mathbf{R} and stretch \mathbf{U} tensors are distinct from those that arise from a polar decomposition of a deformation gradient, namely \mathbf{R} and \mathbf{U} , as found in any, modern, continuum mechanics text. McLellan [46, 47] introduced the Laplace stretch in 1976, which he denoted as \mathbf{H} , while Srinivasa [42] denoted it as $\tilde{\mathbf{F}}$ in his 2012 paper.

An orthogonal tensor \mathcal{R} rotates the reference co-ordinate axes (\vec{e}_1, \vec{e}_2) into a physical co-ordinate system $(\tilde{\vec{e}}_1, \tilde{\vec{e}}_2)$ through an angle θ , which is the fourth physical attribute arising from a **QR** factorization of \mathbf{F} . This angle of rotation describes an orthogonal matrix whereby

$$[\mathcal{R}] = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \quad \text{where} \quad \mathcal{R} = \tilde{\vec{e}}_i \otimes \vec{e}_j = \mathcal{R}_{ij} \vec{e}_i \otimes \vec{e}_j \quad (3.18)$$

with

$$\sin \theta = \frac{F_{21}}{\sqrt{F_{11}^2 + F_{21}^2}}, \quad \cos \theta = \frac{F_{11}}{\sqrt{F_{11}^2 + F_{21}^2}} \quad \therefore \quad \theta = \tan^{-1} \left(\frac{F_{21}}{F_{11}} \right) \quad (3.19)$$

and as such $[\{\tilde{\vec{e}}_1\}\{\tilde{\vec{e}}_2\}] = [\{\vec{e}_1\}\{\vec{e}_2\}][\mathcal{R}]$. A positive angle θ corresponds with a clockwise rotation of physical axes $(\tilde{\vec{e}}_1, \tilde{\vec{e}}_2)$ about reference axes (\vec{e}_1, \vec{e}_2) .

3.2.5.2. Dilemma

There has been a tacit assumption in prior applications of the Gram-Schmidt factorization of \mathbf{F} that the physical base vectors $(\tilde{\vec{e}}_1, \tilde{\vec{e}}_2)$ always satisfy a geometric condition whereby the physical 1-direction $\tilde{\vec{e}}_1$ rotates out of the reference 1-direction \vec{e}_1 , but this need not always be the case. Physical vector $\tilde{\vec{e}}_1$ could equally likely rotate out of the 2-direction \vec{e}_2 of the reference frame. At issue is not how the physical base vectors orient in space. That is managed by Gram's procedure. Rather, at issue is: How do the physical base vectors index with respect to the reference base vectors?

To illustrate this concern, consider two deformation histories, as drawn in Fig. 3.3, each of which describes a simple shear taking place in the plane of a membrane. In one case shear occurs in the 1-direction, while in the other case shear occurs in the 2-direction. These motions lead to Gram-Schmidt factorizations of the deformation gradient, when following the protocol of Eqns. (3.16–3.19), that are described by

$$[\mathbf{F}] = \begin{bmatrix} 1 & \gamma \\ 0 & 1 \end{bmatrix} \implies [\mathcal{R}] = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad [\mathcal{U}] = \begin{bmatrix} 1 & \gamma \\ 0 & 1 \end{bmatrix} \quad (3.20a)$$

and

$$[\mathbf{F}] = \begin{bmatrix} 1 & 0 \\ \gamma & 1 \end{bmatrix} \implies \begin{cases} [\mathcal{R}] = \frac{1}{\sqrt{1+\gamma^2}} \begin{bmatrix} 1 & -\gamma \\ \gamma & 1 \end{bmatrix} \\ [\mathcal{U}] = \begin{bmatrix} \sqrt{1+\gamma^2} & \gamma \\ 0 & 1/\sqrt{1+\gamma^2} \end{bmatrix} \end{cases} \quad (3.20b)$$

where we see that shear \mathcal{U}_{12} has the same physical interpretation in both cases, viz., γ , but the elongations \mathcal{U}_{11} and \mathcal{U}_{22} do not, viz., $\mathcal{U}_{11} = 1$ and $\mathcal{U}_{22} = 1$ in Eq. (3.20a), whereas $\mathcal{U}_{11} = \sqrt{1+\gamma^2}$ and $\mathcal{U}_{22} = 1/\sqrt{1+\gamma^2}$ for the motion described in Eq. (3.20b). Consequently, two geometric interpretations are produced for just one physical mode of deformation. This cannot be!

The only difference between the motions that lead to the two deformation gradients presented in Eq. (3.20) is our choice for labeling the co-ordinate directions. Matrix operations of row and column pivoting, taken from linear algebra, allow one to transform the

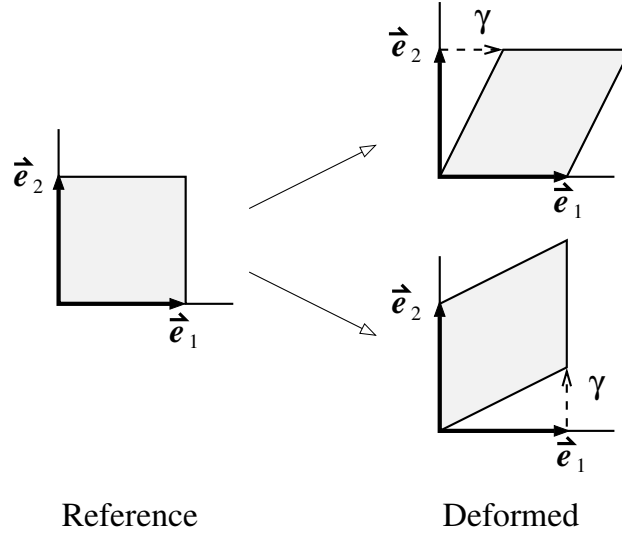


Figure 3.3: The left graphic designates a reference configuration while the right two graphics designate deformed configurations, both in basis (\vec{e}_1, \vec{e}_2) . The top graphic associates with the motion of Eqn. (3.20a), while the bottom graphic associates with the motion of Eqn. (3.20b).

lower-triangular form of Eq. (3.20b) into an upper-triangular form like Eq. (3.20a); hence, producing an unified physical interpretation for both shearing motions, and thereby providing a means for establishing a remedy to this dilemma.

3.2.5.3. Remedy

For 2D membranes, there are only two co-ordinate re-indexings that are possible (for 3D solids there are six). The default is that there is no re-indexing at all, in which case

$$\mathbf{Q} = \mathbf{Q}_0 := \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \implies \begin{bmatrix} \mathcal{F}_{11} & \mathcal{F}_{12} \\ \mathcal{F}_{21} & \mathcal{F}_{22} \end{bmatrix} := \begin{bmatrix} F_{11} & F_{12} \\ F_{21} & F_{22} \end{bmatrix} \quad (3.21a)$$

while in the second case there is a re-indexing specified by

$$\mathbf{Q} = \mathbf{Q}_1 := \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \implies \begin{bmatrix} \mathcal{F}_{11} & \mathcal{F}_{12} \\ \mathcal{F}_{21} & \mathcal{F}_{22} \end{bmatrix} := \begin{bmatrix} F_{22} & F_{21} \\ F_{12} & F_{11} \end{bmatrix} \quad (3.21b)$$

where components $\mathcal{F}_{ij} = Q_{ik}F_{kl}Q_{jl}$ are the ones to be used in Gram-Schmidt factorization, and where $\mathbf{Q} \in \{\mathbf{Q}_0, \mathbf{Q}_1\}$ is orthogonal, i.e., $\mathbf{Q}\mathbf{Q}^\top = \mathbf{Q}^\top\mathbf{Q} = \mathbf{I}$ with $\det \mathbf{Q} = \pm 1$; specifically, $\det \mathbf{Q}_0 = +1$ while $\det \mathbf{Q}_1 = -1$.

The challenge in implementing such a strategy is to determine when to switch from \mathbf{Q}_0 (case 1) to \mathbf{Q}_1 (case 2), or back again, viz., from \mathbf{Q}_1 to \mathbf{Q}_0 . Continuity in the physical fields of deformation (a, b, g) must be satisfied in order for such a change in co-ordinate frame to be physically meaningful. To this end, it is useful to represent the components of a planar

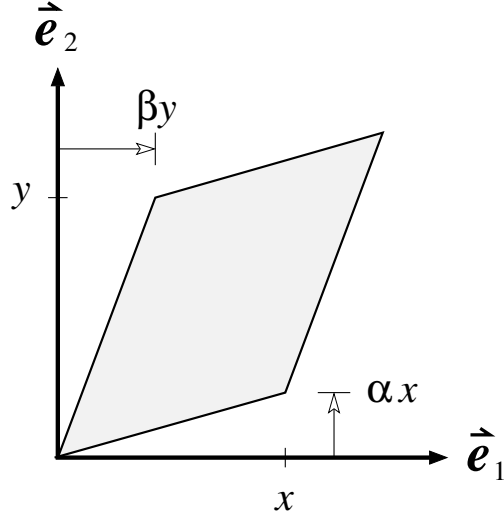


Figure 3.4: A general description of an uniform planar deformation, where $x, y \in \mathbb{R}_+$ and $\alpha, \beta \in \mathbb{R}$. Shears α and β are drawn in their positive sense.

deformation gradient as

$$\begin{bmatrix} \mathcal{F}_{11} & \mathcal{F}_{12} \\ \mathcal{F}_{21} & \mathcal{F}_{22} \end{bmatrix} = \begin{cases} \text{case 1 :} & \begin{bmatrix} F_{11} & F_{12} \\ F_{21} & F_{22} \end{bmatrix} = \begin{bmatrix} x & \beta y \\ \alpha x & y \end{bmatrix} \\ \text{case 2 :} & \begin{bmatrix} F_{22} & F_{21} \\ F_{12} & F_{11} \end{bmatrix} = \begin{bmatrix} y & \alpha x \\ \beta y & x \end{bmatrix} \end{cases} \quad (3.22)$$

where $x = F_{11}$ and $y = F_{22}$ are elongations, while ratios $\alpha = F_{21}/F_{11}$ and $\beta = F_{12}/F_{22}$ are magnitudes of shear, as illustrated in Fig. 3.4.

The physical attributes for Laplace stretch, as they pertain to the two cases in Eqn. (3.21), written in terms of components F_{ij} from $\mathbf{F} = F_{ij} \vec{e}_i \otimes \vec{e}_j$ as defined in Eqn. (3.22), are respectively given by

$$\tilde{a} = x\sqrt{1 + \alpha^2} \quad \hat{a} = y\sqrt{1 + \beta^2} \quad (3.23a)$$

$$\tilde{b} = y(1 - \alpha\beta) / \sqrt{1 + \alpha^2} \quad \hat{b} = x(1 - \alpha\beta) / \sqrt{1 + \beta^2} \quad (3.23b)$$

$$\tilde{g} = y(\alpha + \beta) / x(1 + \alpha^2) \quad \hat{g} = x(\alpha + \beta) / y(1 + \beta^2) \quad (3.23c)$$

$$\tilde{\theta} = \tan^{-1}(-\alpha) \quad \hat{\theta} = \tan^{-1}(-\beta) \quad (3.23d)$$

where attributes in the left column apply to case 1 (i.e., Eqn. 3.21a) while those in the right column apply to case 2 (viz., Eqn. 3.21b). The set of physical attributes $\{a, b, g, \theta\}$ that are to be used when quantifying Laplace stretch and its inverse, according to Eqn. (3.17), are then selected via the strategy

$$\text{if } |\tilde{g}| \geq |\hat{g}| : \quad \{\tilde{a}, \tilde{b}, \tilde{g}, \tilde{\theta}\} \mapsto \{a, b, g, \theta\} \quad (3.24a)$$

$$\text{else } |\hat{g}| \geq |\tilde{g}| : \quad \{\hat{a}, \hat{b}, \hat{g}, \hat{\theta}\} \mapsto \{a, b, g, \theta\} \quad (3.24b)$$

where it is easily verified that $\tilde{a} = \hat{a}$ and $\tilde{b} = \hat{b}$ whenever $\tilde{g} = \hat{g}$; consequently, the physical attributes of deformation a, b, g remain continuous across a co-ordinate switch, however, the angle of co-ordinate rotation θ need not remain continuous across such a switch between co-ordinate frames, as they represent rotations out of different co-ordinate directions.

The above strategy returns matrices for the rotation and Laplace stretch described in Eqn. (3.20a) for both deformation gradients presented in Eqn. (3.20). The dilemma is remedied. Laplace stretch, as remedied, has an unique physical interpretation.

There are three kinematic variables that describe a planar membrane: elongations a and b and shear g . These variables may vary both temporally and spatially. The fields introduced above are exported from module `membranes` outlined in Appendix F.2.

3.3. 3D Irregular Dodecahedra

The kinematic variables needed to describe the deformation of an irregular dodecahedron used as an alveolar model are the volume V (see §2.1.3) and its differential change dV , with the former following from Eq. (2.11) and the latter via a suitable finite-difference formula. Because the material contained within an alveolar sac is typically a gas (else a fluid), no further breakdown of the kinematics is deemed necessary here.

3.4. Code Verification: Kinematics

The thermodynamic conjugate pairs of Freed *et al.* [44, 45, 51] result in the following geometric/thermodynamic strain measures for our dodecahedral model: for 1D rods, an axial strain $e = \ln(L/L_0)$; for 2D membranes, a dilation $\xi = \ln \sqrt{ab/a_0b_0} = \ln \sqrt{A/A_0}$, a squeeze (or pure shear) $\varepsilon = \ln \sqrt{ab_0/a_0b} = \ln \sqrt{\Gamma/\Gamma_0}$, and a (simple) shear $\gamma = g - g_0$; and for 3D dodecahedra, a dilatation $\Xi = \ln \sqrt[3]{V/V_0}$.

3.4.1. Isotropic Motions

Imposing the uniform far-field motion of a volumetric expansion onto our dodecahedral model results in a dodecahedral dilatation ($\Xi := \ln \sqrt[3]{V/V_0}$) that equals its pentagonal dilation ($\xi := \ln \sqrt{A/A_0}$) that equals its chordal strain ($e := \ln(L/L_0)$). These three strain measures follow from the 3-mode thermodynamic theory of Freed *et al.* [44, 45]. Other choices for strain measures do not result in one-to-one relationships when exposed to an isotropic motion like those observed here. This is a particularly useful result in that it establishes a meaningful scaling in terms of strains between the three dimensions, cf. Fig. 3.5. It also provides for a verification of the numerical implementation of our dodecahedral model.

3.4.1.1. Geometric vs. Thermodynamic Strains

There are two types of strain measures that one can use to quantify deformation within a pentagon of a dodecahedron: geometric and thermodynamic. For the uniform far-field motion of volumetric expansion, only the thermodynamic strain known as dilation, i.e., $\xi = \ln \sqrt{ab/a_0b_0}$, varies with the motion, and its response equals that of the geometric

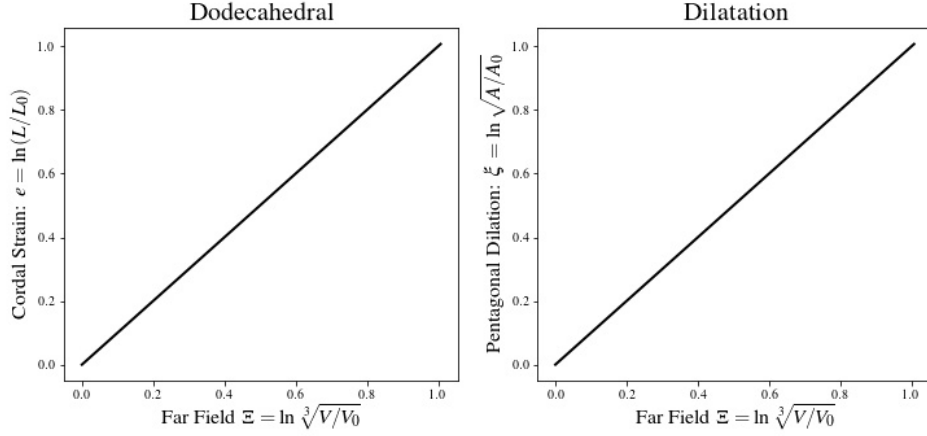


Figure 3.5: Response of a dodecahedron exposed to an isotropic motion of dilatation. The abscissa is the control variable and the ordinates are response variables. The right graphic plots the areal response of the pentagons $\xi = \ln \sqrt{A/A_0}$, while the left graphic plots the axial response of the chords $e = \ln(L/L_0)$. Both are plotted against the volumetric response of the dodecahedron $\Xi = \ln \sqrt[3]{V/V_0}$. Here V denotes dodecahedral volume, A denotes pentagonal area, and L denotes chordal length, all being evaluated in the current state, whose reference values are V_0 , A_0 and L_0 .

strain $\ln \sqrt{A/A_0}$, see Fig. 3.6. Also present in this graph is an observation that the thermodynamic strains for squeeze ε and shear γ do not contribute under motions of pure dilatation, as expected. This further verifies the numerical implementation of our dodecahedral model.

To put this into perspective, we compare with studies done by multiple investigators where ratios of alveolar surface area, viz., A/A_0 , have been measured in rat, rabbit, guinea pig, and cat, cf. Roan & Waters [13, Table 1]. These experiments considered ranges that went as low as 25% and as high as 100% of total lung capacity. Taking statistics of their tabulation produced results of: $A/A_0 = 1.47 \pm 0.44$ during inflation and $A/A_0 = 1.18 \pm 0.14$ during deflation, which correspond to a $\xi = \ln \sqrt{A/A_0} = 0.19 \pm 0.18$ for inflation and a $\xi = \ln \sqrt{A/A_0} = 0.08 \pm 0.07$ for deflation. These areal strains values coincide with the chordal strains of $e = \ln(L/L_0) = 0.13$ measured in vivo around the periphery of an alveolus in rat lung, as reported by Perlman & Bhattachary [52]. Our kinematics have been verified well past these physiologic ranges, viz., for dilatations up to 100% strain.

3.4.2. Isochoric Motions

The motions of pure and simple shears are volume preserving. Imposing these shears as far-field motions onto our dodecahedral model produced the results displayed in Fig. 3.7. For simple shear, the numerical model is in error by about machine precision, i.e., $\epsilon_m \approx 2.2 \times 10^{-16}$, for strains up to 100%, while for pure shear (a special case of squeeze in 3D) the model is in error by about machine precision for strains up to of about 60%, after which the error further increases, up to about $10\epsilon_m$ at strains around 100%. This further verifies the numerical implementation of our dodecahedral model.

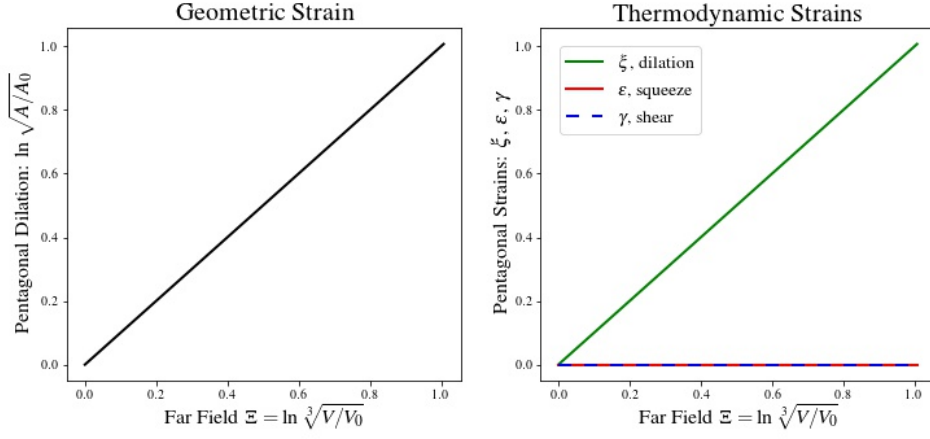


Figure 3.6: Response of a dodecahedron exposed to a far-field isotropic motion of dilatation. The abscissa is the control variable and the ordinates are response variables. The right graphic plots the three thermodynamic strains, as they apply to a pentagon, while the left graphic plots the geometric strain of a pentagon.

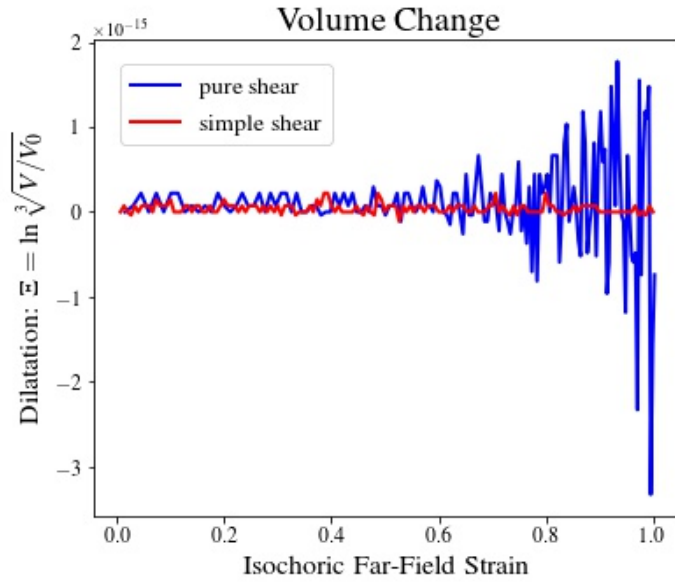


Figure 3.7: Response of a dodecahedron exposed to far-field motions of pure and simple shears. Note that the ordinate is $\times 10^{-15}$ and machine precision is $\sim 2.2 \times 10^{-16}$.

3.4.2.1. Geometric Strains

How the thirty chords and the twelve irregular pentagons deform under far-field motions of pure shear is displayed in Fig. 3.8. Figure 3.7 demonstrates that the overall response of a dodecahedron is isochoric during pure shear. Regardless, Fig. 3.8 demonstrates that the individual chordal and pentagonal constituents deform in a non-homogeneous manner, where the strains have been calculated as geometric changes in dodecahedral shape. This result agrees with in vivo observations made by Perlman & Bhattacharya [52] where confocal microscopy was used to image a breathing rat lung.

For the dodecahedral chords, there are six independent responses for motions of pure shear: two chords each for three of these lines, and eight chords each for the remaining three curves present in the left images of Fig. 3.8. For pentagons, there are three independent responses with four pentagons responding according to each curve shown in the right images. Although different chords and pentagons deform differently when sheared in different directions, their collective responses are the same regardless of the far-field direction being sheared. Consequently, the local geometric response of a dodecahedron is isotropic under the far-field motions of pure shear.

How the thirty chords and the twelve irregular pentagons deform under far-field motions of simple shear is displayed in Fig. 3.9. Figure 3.7 demonstrates that the overall response of a dodecahedron is isochoric during a far-field simple shear. Figure 3.9 demonstrates that the individual chordal and pentagonal constituents deform in a non-homogeneous manner during simple shears, like they do for pure shears. However, unlike pure shears whose collective chordal and pentagonal responses remain isotropic, here they diverge from isotropy under motions of simple shear. Simple shears in the 1-2 and 2-3 planes have the same collective response; whereas, simple shear in the 1-3 plane has a slightly different response.

Figures 3.5–3.9 establish that a dodecahedron is (nearly, but not completely) isotropic in its kinematic response, as measured by the geometric strains $e = \ln(L/L_0)$, $\xi = \ln \sqrt{A/A_0}$ and $\Xi = \ln \sqrt[3]{V/V_0}$. Furthermore, even though a far-field deformation is homogeneous, in accordance with Conjecture 1, the local deformations within the individual constituents of an alveolus will typically be heterogeneous, which agrees with imaging data [52].

3.4.2.2. Thermodynamic Strains

Addressing the septal response, modeled here as a set of twelve irregular pentagons per alveolus, we desire to come to a determination regarding how to best model the deformation occurring within these alveolar septa. In the section above we investigated the geometric response of alveolar septa via the strain measure $\ln \sqrt{A/A_0}$, which quantifies dilation.

The thermodynamic strains arising from a Gram-Schmidt factorization of the deformation gradient put forward in §3.2.5 specify three strain measures pertinent to a membrane: dilation $\xi = \ln \sqrt{ab}$, squeeze $\varepsilon = \ln \sqrt{a/b}$ and shear $\gamma = g$, where elongations a and b and magnitude of shear g are illustrated in Fig. 3.2. Of these, dilation is a uniform response, while squeeze and shear describe non-uniform responses. To acquire them requires knowing the deformation gradient.

The curves in Figs. 3.8 & 3.9 were obtained from geometric measures for chordal strain $\ln(L/L_0)$ and areal dilation $\ln \sqrt{A/A_0}$. They were computed under separate conditions of

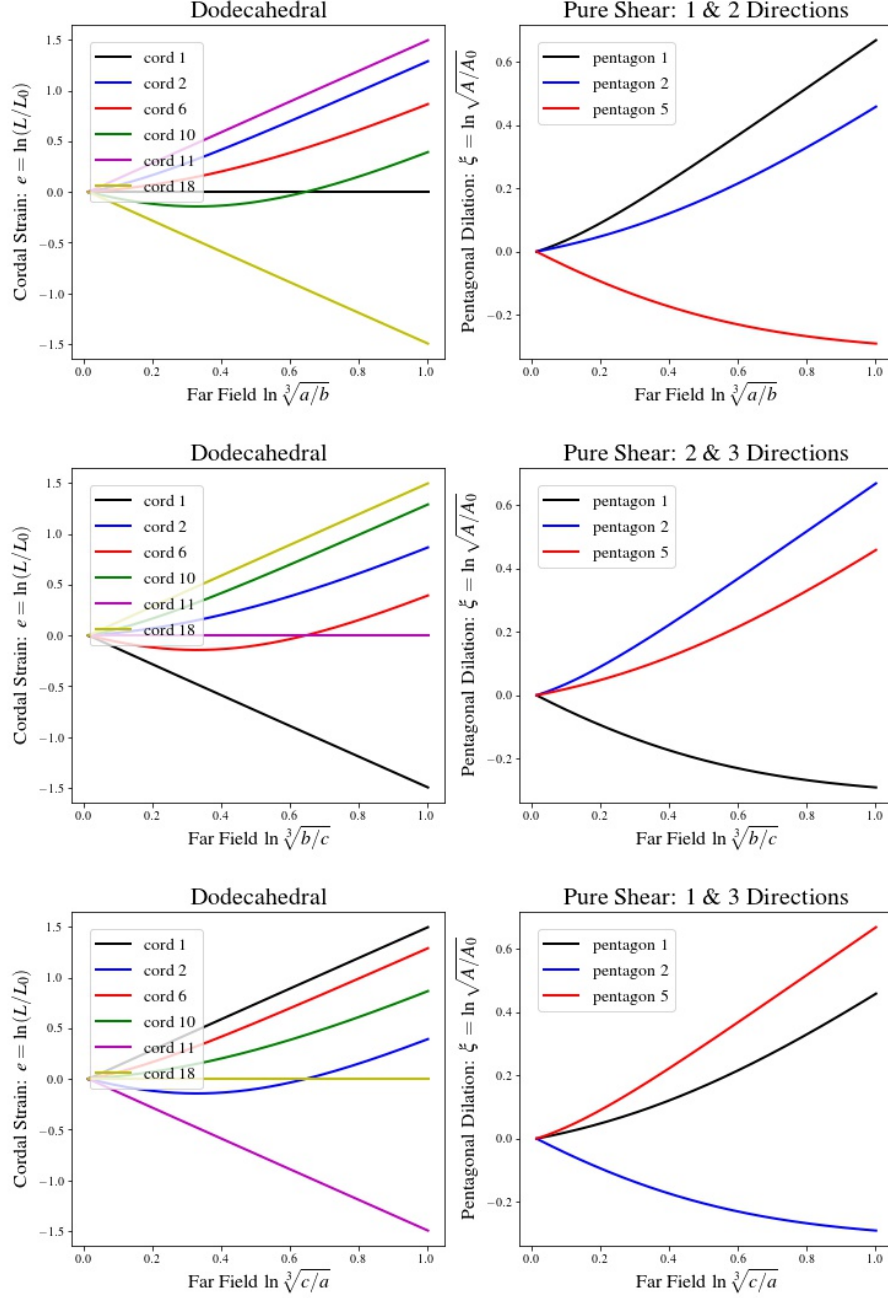


Figure 3.8: Response of a dodecahedron exposed to far-field pure-shear motions in the sense of Treloar [53]: $a = \ell$, $b = 1/\ell$ and $c = 1$ in the top images; $a = 1$, $b = \ell$ and $c = 1/\ell$ in the middle images; and $a = 1/\ell$, $b = 1$ and $c = \ell$ in the bottom images, with ℓ denoting an elongation of extrusion. In all six graphic images, the relevant (controlled) motion of the far-field pure shear is plotted along the abscissa. In each image pair, the right graphic presents pentagonal dilations, while the left graphic presents chordal elongations. Only unique responses are plotted; repetitions are not.

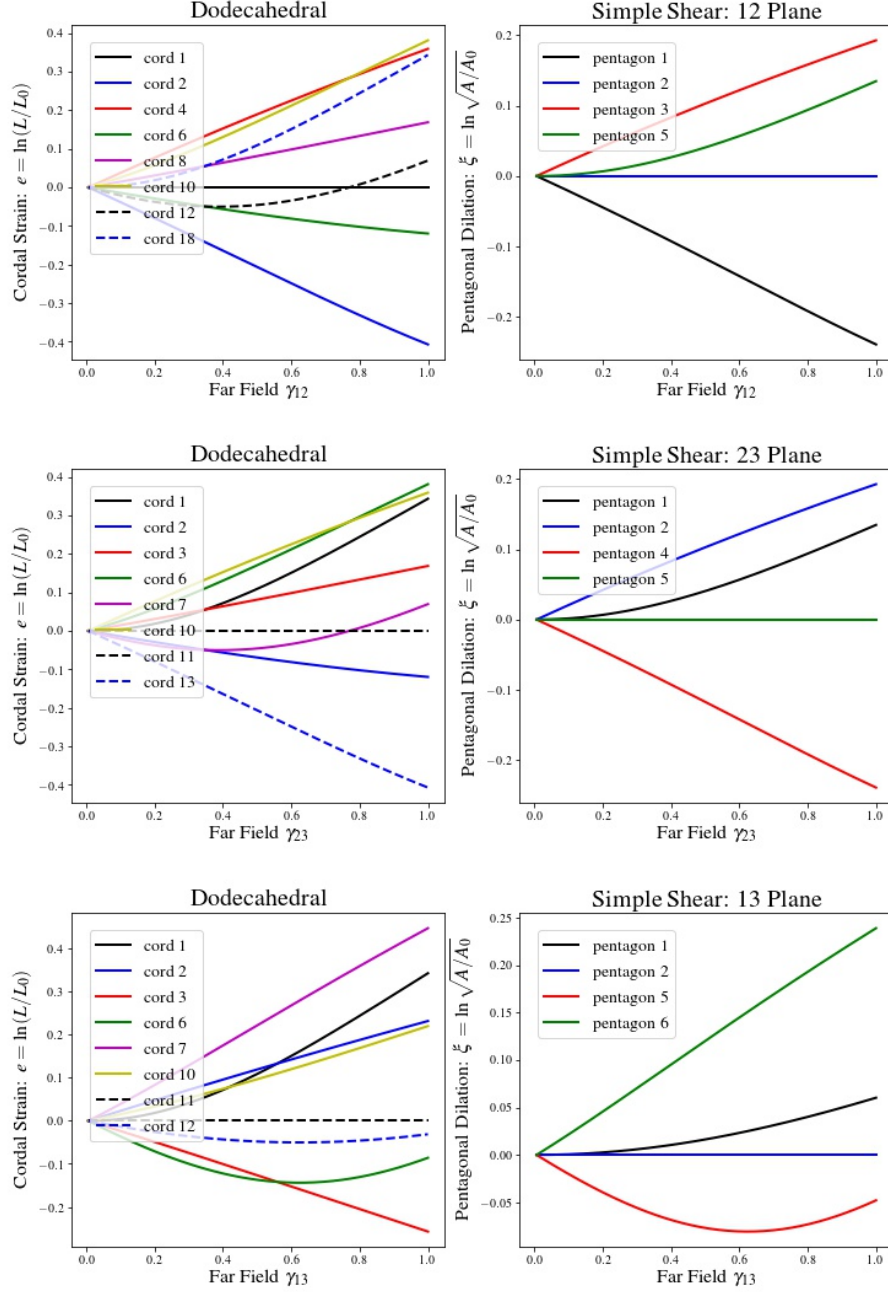


Figure 3.9: Response of a dodecahedron exposed to far-field simple-shear motions. In all six graphic images, the relevant (controlled) motion of simple shear is plotted along the abscissa. In each image pair, the right graphic presents pentagonal dilations, while the left graphic presents chordal elongations. Only unique responses are plotted; repetitions are not.

pure and simple far-field shears. The curves in Figs. 3.10 & 3.11 were obtained from thermodynamic measures for membrane strain under the same far-field deformations. The strains of dilation ξ , squeeze ε , and shear γ were computed in accordance with §3.2.5 using deformation gradients gotten from the pentagonal shape functions of Wachspress [36] discussed in §3.2.1.⁵

Figures 3.8–3.11 allow us to conclude that if septal dilation were the only mode of planar deformation thought to cause a mechanical response, then knowledge of the geometric strain $\xi = \ln \sqrt{A/A_0}$ would be adequate; there would be no need to introduce a separate finite-element discretization of the septal planes for acquiring their deformation gradients. However, if the non-uniform responses of squeeze ε and shear γ are thought to contribute to the overall mechanical response of these membranes, then the shape functions of Wachspress [36, 37] ought to be used for acquiring the deformation gradient within a septal plane, as constant-strain triangles are not accurate enough for this application. Strains derived from Wachspress shape functions are inhomogeneous; consequently, the deformation gradient will need to be evaluated at each Gauss point of integration within a pentagon, cf. §5.2.

3.4.3. Co-ordinate Pivoting

The pivoting strategy of §3.2.5.3 used to address the physical dilemma of §3.2.5.2 did not engage often during our assessment of the code, but it did arise at least twice with effects illustrated in Figs. 3.12 & 3.13. Here one can see that there is a clear effect on the shear response within four pentagonal planes; however, no change is observed to have occurred in either the dilation or squeeze responses, as expected. It is not always possible to know when or where a co-ordinate relabeling ought to occur; consequently, the algorithm put forward in §3.2.5.3 is deemed necessary.

3.4.4. Compatible Membrane Deformations

For a deformation to be compatible, and therefore integrable, the curl of its deformation gradient must vanish, viz., $\text{curl}(\mathbf{F}) = \mathbf{0}$ [41]. Equation (3.12) provides constraint equations for the compatibility of planar motions, e.g., septal planes of an alveolus. Here we test to make sure that these conditions are satisfied within the pentagonal planes of our alveolar dodecahedron, assuming that the shape functions of Wachspress apply.

Figure 3.14 presents the compatibility response at the centroid of a typical pentagonal plane during the uniform expansion of a regular dodecahedron out to 100% strain. Theoretically, all four derivatives should be zero for this motion. Actually, their values are on the order of machine precision. Most importantly, whenever they are not zero, they lie along the 45° diagonal, thereby verifying compatibility in the case of a dilatation.

Similarly, Figs. 3.15 & 3.16 present typical responses for testing compatibility during far-field pure shear (Fig. 3.15) and simple shear (Fig. 3.16) deformations. In both cases, one of the four pentagons around the girth of the dodecahedron (viz., #5) has been selected, as both modes of deformation are activated in this pentagon. In both cases, errors are typically

⁵Five constant-strain triangles were also used to quantify the deformation gradient for each pentagonal surface at its centroid—the common vertex to all five triangles. This approach provided accurate descriptions for uniform strain, i.e., dilation ξ , but not for the two non-uniform strains, viz., squeeze ε and shear γ .

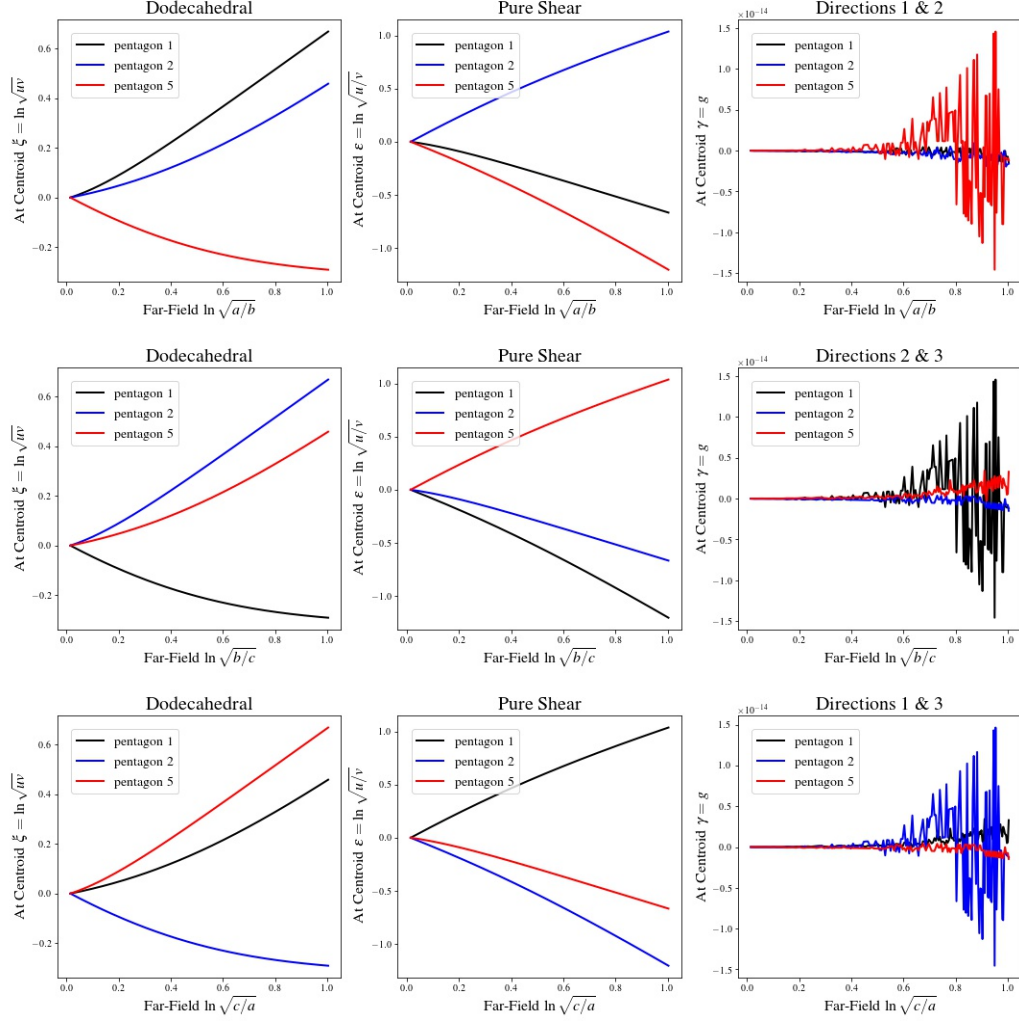


Figure 3.10: Same boundary conditions as in Fig. 3.8. Pentagonal areas were used to compute dilation in Fig. 3.8. The shape functions of Wachspress were used to compute dilation here. The uniform response in the right column of Fig. 3.8 and in the left column above are the same, providing additional assurance that the code has been correctly implemented. The squeeze response shown in the center column is the same for all three orientations of far-field pure shear, i.e., this response is isotropic. The right column has ordinates scaled by 10^{-14} implicating that there is no simple shear response occurring within any pentagonal surface of the dodecahedron whenever it is subjected to a far-field motion of pure shear.

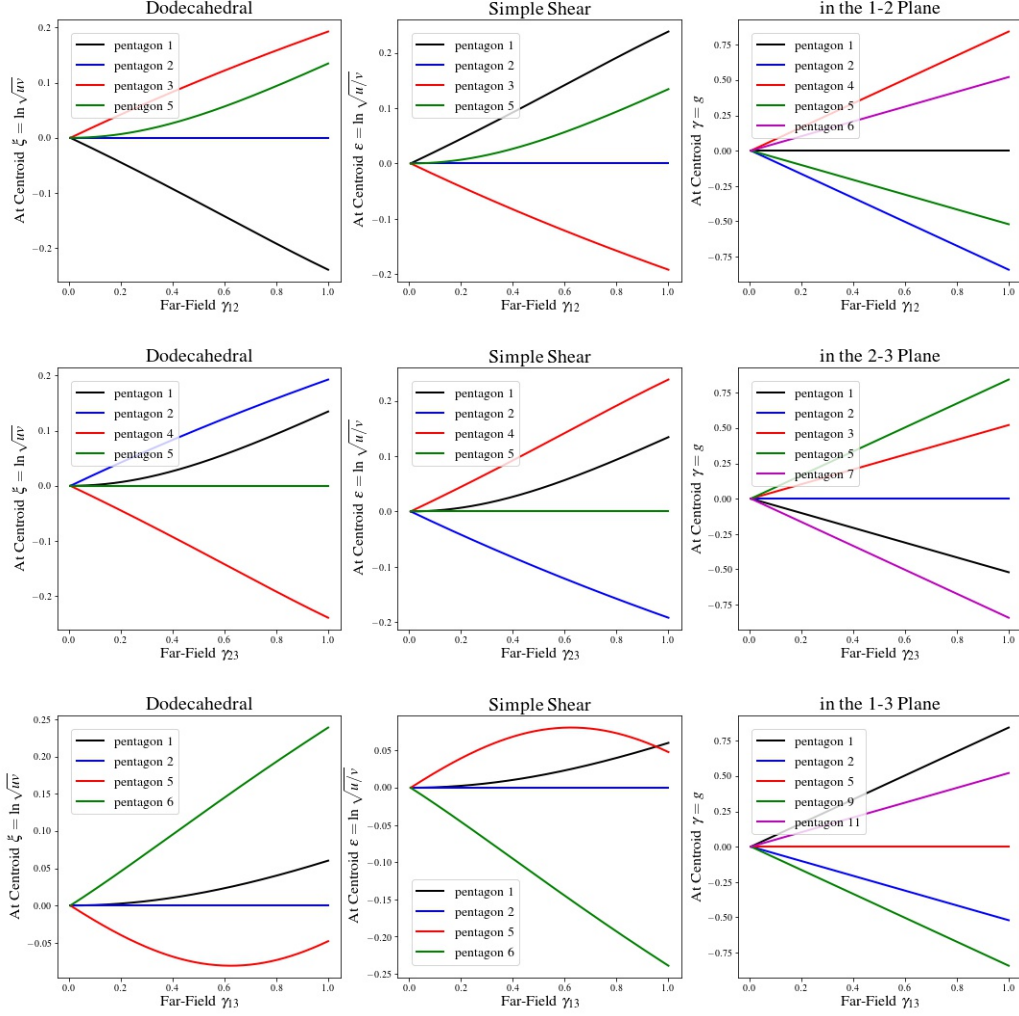


Figure 3.11: Same boundary conditions as in Fig. 3.9. Pentagonal areas were used to compute dilation in Fig. 3.9. The shape functions of Wachspress were used to compute dilation here. The uniform response in the right column of Fig. 3.9 and in the left column above are the same, providing additional assurance that the code has been correctly implemented. Like the dilational responses of the left column, the squeeze responses of the center column are the same in the 1-2 and 2-3 planes, but differ in the 1-3 plane. In all cases, the simple shear response of any pentagonal plane is proportional to that of the far-field shear imposed, further substantiating the code's implementation. The shear response of the septal membranes is isotropic.

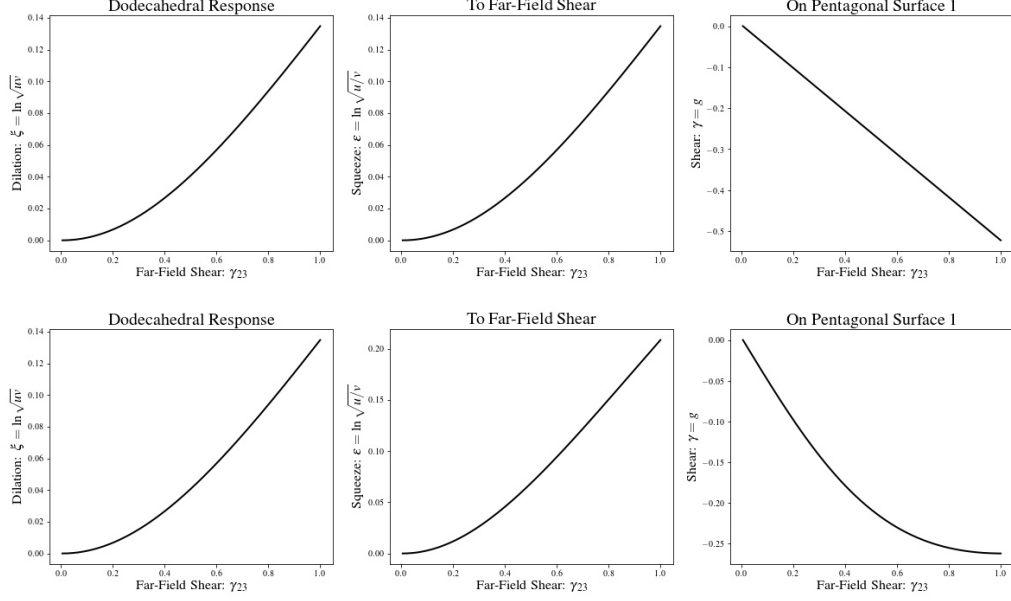


Figure 3.12: A far-field shear of γ_{23} is imposed on the dodecahedron. Pentagons 1 and 8 exhibit the plotted response. The top set of figures result whenever the pivoting strategy of §3.2.5.3 is used, while the bottom set of figures result whenever no pivoting strategy is employed. The dilation (left graphs) and squeeze (center graphs) responses are not affected by pivoting, only shear (right graphs) is affected. Pivoting maintains a linear shear response under a far-field shearing of the dodecahedron, as desired.

less than ten times machine precision, thereby verifying compatibility in the cases of squeeze and shear.

This collective set of graphs, Figs. 3.14–3.16, investigate the constraint of compatibility in terms of the three fundamental modes of deformation: dilatation, squeeze and shear. These figures verify that the constraint of compatibility is satisfied when using the pentagonal shape functions of Wachspress [36, 37] in our dodecahedral model, as errors are typically less than ten times machine precision. This has been verified out to deformations that are at least three times those of their normal physiologic range.

Our kinematic analysis of a dodecahedron has been verified theoretically and numerically.

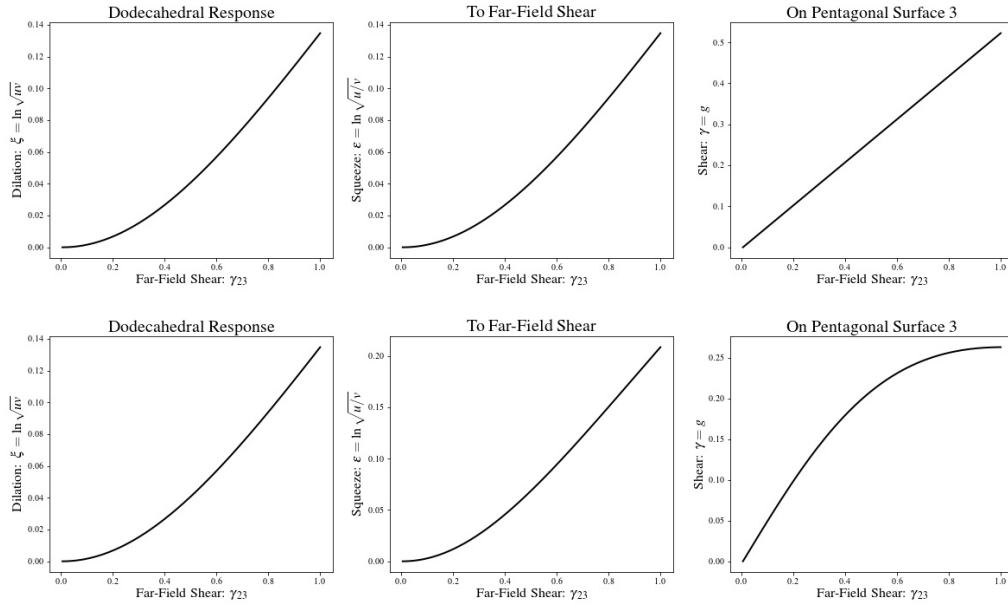


Figure 3.13: A far-field shear of γ_{23} is imposed on the dodecahedron. Pentagons 3 and 10 exhibit the plotted response. The top set of figures result whenever the pivoting strategy of §3.2.5.3 is used, while the bottom set of figures result whenever no pivoting strategy is employed. The dilation (left graphs) and squeeze (center graphs) responses are not affected by pivoting, only shear (right graphs) is affected. Pivoting maintains a linear shear response under a far-field shearing of the dodecahedron, as desired.

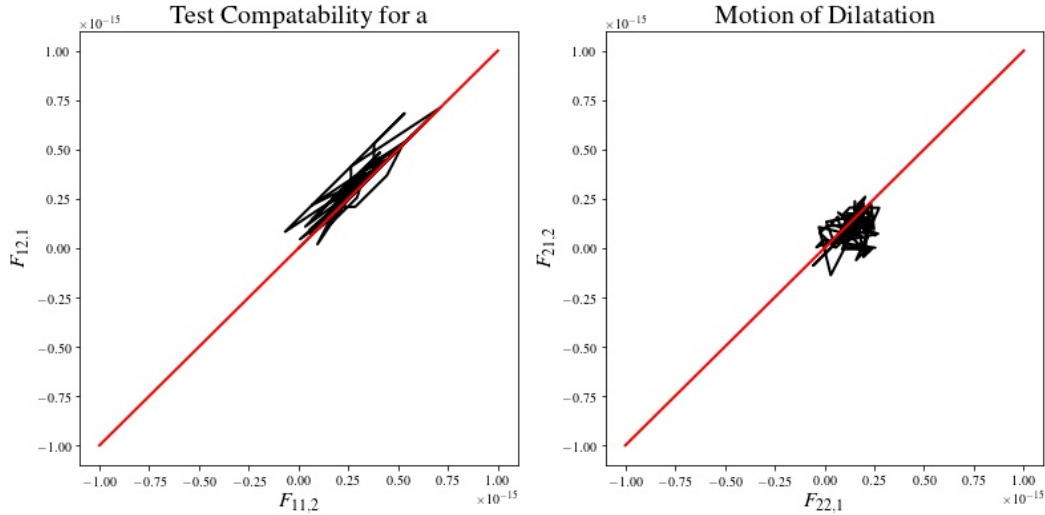


Figure 3.14: Planar compatibility requires $F_{11,2} = F_{12,1}$ and $F_{22,1} = F_{21,2}$ where the left-hand sides are plotted as the abscissæ and the right-hand sides are plotted as the ordinates. For compatibility, the response ought to lie along the 45° diagonal, which is drawn in red over the range of $\pm 10^{-15}$ where machine precision is about 2.2×10^{-16} . Here the motion is one of uniform dilatation out to 100% strain.

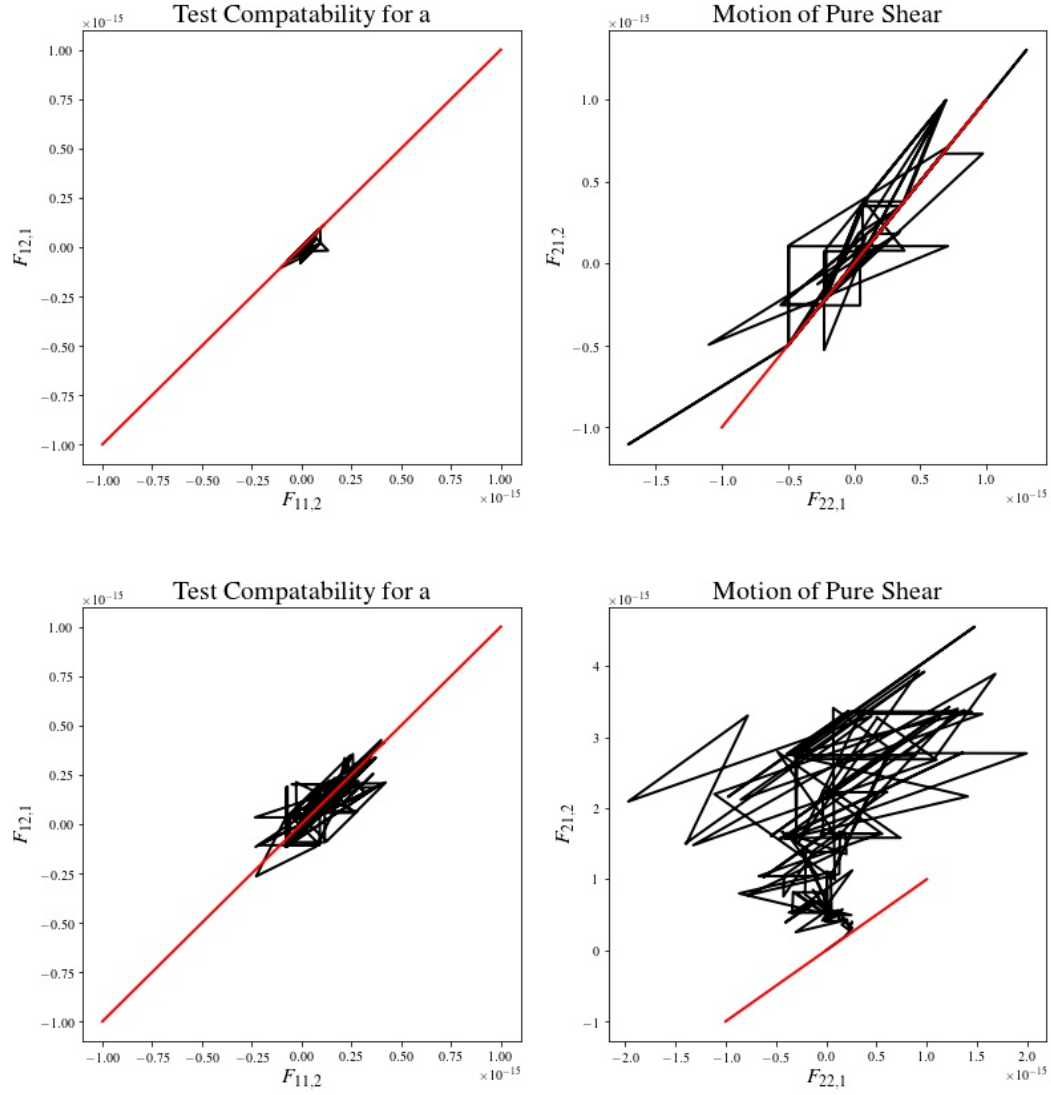


Figure 3.15: Planar compatibility requires $F_{11,2} = F_{12,1}$ and $F_{22,1} = F_{21,2}$ where the left-hand sides are plotted as the abscissæ and the right-hand sides are plotted as the ordinates. For compatibility, the response ought to lie along the 45° diagonal, which is drawn in red over the range of $\pm 10^{-15}$ where machine precision is about 2.2×10^{-16} . Here the motion is one of pure shear out to 100% strain with elongation occurring in the 1-direction, contraction occurring in the 2 direction, while the 3-direction is held fixed. These results pertain to pentagon 5: nodes 15, 5, 12, 11, 1, cf. Fig. 2.1 and Table 2.3. The top row of figures is the best response (at Gauss point 7, cf. Fig. 5.1) while the bottom row of figures is the worst response (at Gauss point 5).

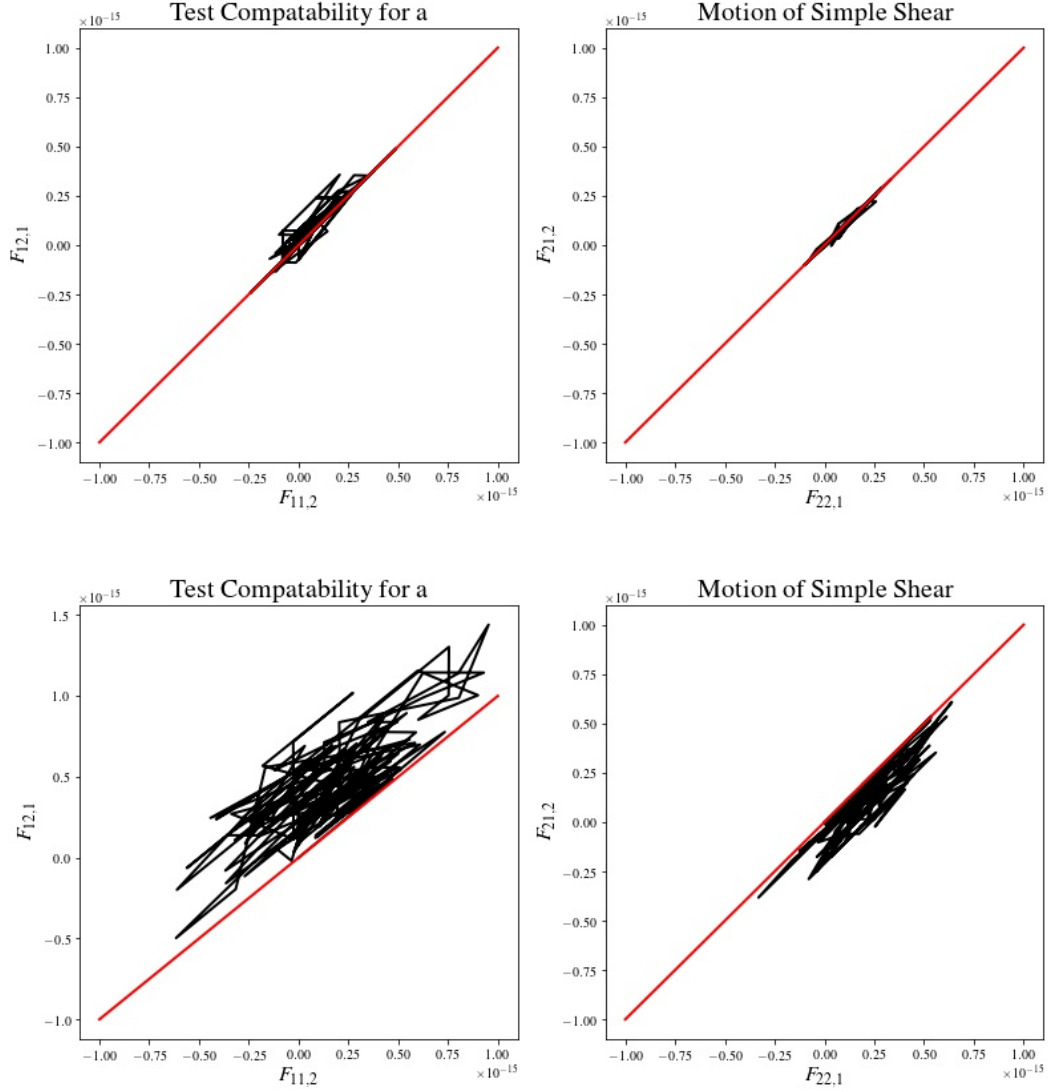


Figure 3.16: Planar compatibility requires $F_{11,2} = F_{12,1}$ and $F_{22,1} = F_{21,2}$ where the left-hand sides are plotted as the abscissæ and the right-hand sides are plotted as the ordinates. For compatibility, the response ought to lie along the 45° diagonal, which is drawn in red over the range of $\pm 10^{-15}$ where machine precision is about 2.2×10^{-16} . Here the motion is one of simple shear out to 100% strain, shearing along 1-2 planes in the 1-direction. These results pertain to pentagon 5: nodes 15, 5, 12, 11, 1, cf. Fig. 2.1 and Table 2.3. The top row of figures is the best response (at Gauss point 7, cf. Fig. 5.1) while the bottom row of figures is the worst response (at Gauss point 5).

Part 4

Constitutive Theory

Roan & Waters [13] and Suki *et al.* [54, 55] have each written extensive review articles on the mechanics of parenchyma, have provided detailed information about the structural constituents of alveoli, and have discussed their influence on the overall mechanical response of parenchyma. Of particular importance, from a mechanics perspective, are the constituent building blocks of alveolar tissue: collagen (types I and III, predominantly), elastin, proteoglycans and other proteins, surfactant and cells (epithelial and endothelial, predominantly). These constituents are assembled in such a manner so as to produce a variety of alveolar sub-structures that are essentially 1D (alveolar chords), 2D (alveolar septa) or 3D (alveolar sacs) in their geometric construction.

A dodecahedron is used here as a geometric model for an alveolus [18], cf. Figs. 1.1 & 2.1. It is comprised of: thirty 1D rods that represent alveolar chords, twelve 2D membranes that represent alveolar septa and that are considered to be pentagonal in shape, and one 3D cavity filled with air (or fluids in the case of a contusion caused by injury or an edema caused by disease) that is considered to be dodecahedral in shape. Thermoelastic constitutive equations are derived for these three spatial geometries in this chapter.

We recall from our kinematic study of a dodecahedron that the geometric strains of $e := \ln(L/L_0)$ for the elongation of septal chords, $\xi := \ln \sqrt{A/A_0}$ for the dilation of septal membranes, and $\Xi := \ln \sqrt[3]{V/V_0}$ for the dilatation of alveolar volume are equivalent to one another under motions of uniform expansion. These three, geometric, strain measures also exist as thermodynamic strains, each associating with a distinct and unique conjugate stress.

Constitutive equations are a derived consequence from physical laws governing thermodynamic processes. Here we derive constitutive equations applicable for 1D thermoelastic fibers (alveolar chords), 2D thermoelastic membranes (alveolar septa), and 3D thermoelastic volumes (alveolar sacs). In §4.1, we assume that the motions are uniform in their spatial dimension. Later, in §4.2, the non-uniform motions of shear associated with a membrane are included into this thermodynamic framework. Section 4.3 pulls these results together, sufficient for the intended purpose of modeling the three structural facets that comprise an alveolus.

4.1. Green Thermoelastic Solids Subjected to Uniform Motions in 1D, 2D & 3D

Combining the First and Second Laws of Thermodynamics governing uniform, reversible, adiabatic processes results in the following three formulæ, one per dimension; they are

$$\text{In 1 Dimension:} \quad d\mathcal{U} = \theta d\eta + \frac{1}{\rho_{1D}} F dL/L \quad (4.1a)$$

$$\text{In 2 Dimensions:} \quad d\mathcal{U} = \theta d\eta + \frac{1}{\rho_{2D}} T dA/A \quad (4.1b)$$

$$\text{In 3 Dimensions:} \quad d\mathcal{U} = \theta d\eta - \frac{1}{\rho_{3D}} P dV/V \quad (4.1c)$$

wherein \mathcal{U} is an internal energy density (erg/gr = dyne.cm/gr), which is a function of state, θ is a temperature in Kelvin ($273 + ^\circ\text{C}$), η is an entropy density (erg/gr.K), whereas the mass densities ρ_{1D} (gr/cm), ρ_{2D} (gr/cm²) and ρ_{3D} (gr/cm³) associate with a reference state of per unit length, or per unit area, or per unit volume, as appropriate. Pressure P is assigned to be positive whenever a body undergoes a hydrostatic compression, per accepted practice.

4.1.1. Constitutive Equations

Because the internal energy density \mathcal{U} is a state function, its total derivative produces a Pfaffian form [56] out of which the following constitutive formulæ are readily obtained

$$\text{In 1 Dimension:} \quad \theta = \partial_\eta \mathcal{U}(\eta, e) \quad F = \rho_{1D} \partial_e \mathcal{U}(\eta, e) \quad (4.2a)$$

$$\text{In 2 Dimensions:} \quad \theta = \partial_\eta \mathcal{U}(\eta, \xi) \quad 2T = \rho_{2D} \partial_\xi \mathcal{U}(\eta, \xi) \quad (4.2b)$$

$$\text{In 3 Dimensions:} \quad \theta = \partial_\eta \mathcal{U}(\eta, \Xi) \quad -3P = \rho_{3D} \partial_\Xi \mathcal{U}(\eta, \Xi) \quad (4.2c)$$

where we have used the strain definitions $e := \ln(L/L_0)$, $\xi := \ln \sqrt{A/A_0}$ and $\Xi := \sqrt[3]{V/V_0}$, in accordance with Part 3. These constitutive equations govern Green thermoelastic solids undergoing uniform motions in adiabatic enclosures. As a convenience, we employ the following notation $\partial_\eta \mathcal{U} := \partial \mathcal{U} / \partial \eta$, etc.

Considering the response variables for temperature and force/surface-tension/pressure to be C^1 functions of state (cf. Weinhold [57] and Gilmore [58]) in a Green thermoelastic solid undergoing an uniform adiabatic motion, one can then differentiate Eqn. (4.2) thereby producing the following collection of coupled differential equations

$$\text{In 1 Dimension:} \quad \begin{Bmatrix} d\theta \\ dF \end{Bmatrix} = \begin{bmatrix} \partial_{\eta\eta} \mathcal{U} & \partial_{\eta e} \mathcal{U} \\ \rho_{1D} \partial_{e\eta} \mathcal{U} & \rho_{1D} \partial_{ee} \mathcal{U} \end{bmatrix} \begin{Bmatrix} d\eta \\ de \end{Bmatrix} \quad (4.3a)$$

$$\text{In 2 Dimensions:} \quad \begin{Bmatrix} d\theta \\ 2dT \end{Bmatrix} = \begin{bmatrix} \partial_{\eta\eta} \mathcal{U} & \partial_{\eta\xi} \mathcal{U} \\ \rho_{2D} \partial_{\xi\eta} \mathcal{U} & \rho_{2D} \partial_{\xi\xi} \mathcal{U} \end{bmatrix} \begin{Bmatrix} d\eta \\ d\xi \end{Bmatrix} \quad (4.3b)$$

$$\text{In 3 Dimensions:} \quad \begin{Bmatrix} d\theta \\ -3dP \end{Bmatrix} = \begin{bmatrix} \partial_{\eta\eta} \mathcal{U} & \partial_{\eta\Xi} \mathcal{U} \\ \rho_{3D} \partial_{\Xi\eta} \mathcal{U} & \rho_{3D} \partial_{\Xi\Xi} \mathcal{U} \end{bmatrix} \begin{Bmatrix} d\eta \\ d\Xi \end{Bmatrix} \quad (4.3c)$$

where, from calculus, mixed partial derivatives satisfy $\partial_{e\eta} \mathcal{U} = \partial^2 \mathcal{U} / \partial e \partial \eta = \partial^2 \mathcal{U} / \partial \eta \partial e = \partial_{\eta e} \mathcal{U}$, etc., that in the thermodynamics literature are known as Maxwell's relations.

Changing cause and effect between entropy and temperature in Eqn. (4.3) leads to

$$\begin{Bmatrix} d\eta \\ dF \end{Bmatrix} = \begin{bmatrix} 1/\partial_{\eta\eta} \mathcal{U} & -\partial_{\eta e} \mathcal{U} / \partial_{\eta\eta} \mathcal{U} \\ \rho_{1D} \partial_{e\eta} \mathcal{U} / \partial_{\eta\eta} \mathcal{U} & \rho_{1D} (\partial_{ee} \mathcal{U} - \partial_{e\eta} \mathcal{U} \cdot \partial_{\eta e} \mathcal{U} / \partial_{\eta\eta} \mathcal{U}) \end{bmatrix} \begin{Bmatrix} d\theta \\ de \end{Bmatrix} \quad (4.4a)$$

$$\begin{Bmatrix} d\eta \\ 2dT \end{Bmatrix} = \begin{bmatrix} 1/\partial_{\eta\eta} \mathcal{U} & -\partial_{\eta\xi} \mathcal{U} / \partial_{\eta\eta} \mathcal{U} \\ \rho_{2D} \partial_{\xi\eta} \mathcal{U} / \partial_{\eta\eta} \mathcal{U} & \rho_{2D} (\partial_{\xi\xi} \mathcal{U} - \partial_{\xi\eta} \mathcal{U} \cdot \partial_{\eta\xi} \mathcal{U} / \partial_{\eta\eta} \mathcal{U}) \end{bmatrix} \begin{Bmatrix} d\theta \\ d\xi \end{Bmatrix} \quad (4.4b)$$

$$\begin{Bmatrix} d\eta \\ -3dP \end{Bmatrix} = \begin{bmatrix} 1/\partial_{\eta\eta} \mathcal{U} & -\partial_{\eta\Xi} \mathcal{U} / \partial_{\eta\eta} \mathcal{U} \\ \rho_{3D} \partial_{\Xi\eta} \mathcal{U} / \partial_{\eta\eta} \mathcal{U} & \rho_{3D} (\partial_{\Xi\Xi} \mathcal{U} - \partial_{\Xi\eta} \mathcal{U} \cdot \partial_{\eta\Xi} \mathcal{U} / \partial_{\eta\eta} \mathcal{U}) \end{bmatrix} \begin{Bmatrix} d\theta \\ d\Xi \end{Bmatrix} \quad (4.4c)$$

which are written above in a format that is more useful for our multiscale application. In this process, a localization procedure pulls temperature and strain from the continuum scale

down to the alveolar scale. The alveolar entropy and stress are then determined via the above constitutive equations. Afterwards, a homogenization procedure pushes this alveolar entropy and stress up to the continuum level. We take on the independent variables of a Helmholtz free energy, but not his potential, preferring to use the internal energy potential to ensure a proper incorporation of Maxwell's constraint.

Constitutive equations (4.3 & 4.4) take on the form of hypo-elastic material models [59], which are ideal for numerical implementation using numerical solution techniques like those presented in Part 5.

4.1.2. Material Constants

Experiments are performed for the purpose of characterizing material behavior. In mechanics, we relate measured material constants to gradients and curvatures of thermodynamic potentials out of which material models are created. Experiments are typically done to quantify the following material constants, selected per a material's physical dimension:

$$C_f := \theta \partial_\theta \eta|_F \quad \alpha_f := L^{-1} \partial_\theta L|_F = \partial_\theta e|_F \quad E_\theta := L \partial_L F|_\theta = \partial_e F|_\theta \quad (4.5a)$$

$$C_t := \theta \partial_\theta \eta|_T \quad \alpha_t := A^{-1} \partial_\theta A|_T = 2 \partial_\theta \xi|_T \quad M_\theta := A \partial_A T|_\theta = \frac{1}{2} \partial_\xi T|_\theta \quad (4.5b)$$

$$C_p := \theta \partial_\theta \eta|_P \quad \alpha_p := V^{-1} \partial_\theta V|_P = 3 \partial_\theta \Xi|_P \quad -K_\theta := V \partial_V P|_\theta = \frac{1}{3} \partial_\Xi P|_\theta \quad (4.5c)$$

Herein, the various specific heats C_f , C_t , C_p (erg/gr.K) are, essentially, all equivalent as they are all defined per unit mass, insensitive to dimension; however, the various thermal expansions α_f , α_t , α_p (1/K) are all distinct, as they are each defined with respect to their physical dimension, viz., $\alpha_f := L^{-1} \partial L / \partial \theta|_F$, $\alpha_t := A^{-1} \partial A / \partial \theta|_T$ and $\alpha_p := V^{-1} \partial V / \partial \theta|_P$. Parameter E_θ is a modulus of extension (dyne), parameter M_θ is a modulus of dilation (dyne/cm), and parameter K_θ is a modulus of dilatation (dyne/cm²), a.k.a. the bulk modulus, with each modulus being measured at fixed temperature. Shear moduli are discussed later in §4.2. The above material constants are gradients; they constitute tangents to their associated physical response curves. Consequently, they need not be of constant value throughout state space, like a Hookean material supposes them to be. This is an important characteristic for our application. Here we employ the commonly used notation $\partial_\theta \eta|_F := (\partial \eta / \partial \theta)|_F$, etc.

In terms of the material constants given in Eqn. (4.5), of which there are three per dimension, the internal energy density has the following three curvatures associated with it. For 1D materials:

$$\partial_{\eta\eta} \mathcal{U} = \frac{\rho_{1D} \theta}{\rho_{1D} C_f - E_\theta \alpha_f^2 \theta} \quad (4.6a)$$

$$\partial_{ee} \mathcal{U} = \frac{C_f E_\theta}{\rho_{1D} C_f - E_\theta \alpha_f^2 \theta} \quad (4.6b)$$

$$\partial_{\eta e} \mathcal{U} \equiv \partial_{e\eta} \mathcal{U} = \frac{-E_\theta \alpha_f \theta}{\rho_{1D} C_f - E_\theta \alpha_f^2 \theta} \quad (4.6c)$$

For 2D materials:

$$\partial_{\eta\eta}\mathcal{U} = \frac{\rho_{2D}\theta}{\rho_{2D}C_t - M_\theta\alpha_t^2\theta} \quad (4.6d)$$

$$\partial_{\xi\xi}\mathcal{U} = \frac{4C_tM_\theta}{\rho_{2D}C_t - M_\theta\alpha_t^2\theta} \quad (4.6e)$$

$$\partial_{\eta\xi}\mathcal{U} \equiv \partial_{\xi\eta}\mathcal{U} = \frac{-2M_\theta\alpha_t\theta}{\rho_{2D}C_t - M_\theta\alpha_t^2\theta} \quad (4.6f)$$

For 3D materials (cf. Weinhold [57] and Gilmore [58]):

$$\partial_{\eta\eta}\mathcal{U} = \frac{\rho_{3D}\theta}{\rho_{3D}C_p - K_\theta\alpha_p^2\theta} \quad (4.6g)$$

$$\partial_{\Xi\Xi}\mathcal{U} = \frac{9C_pK_\theta}{\rho_{3D}C_p - K_\theta\alpha_p^2\theta} \quad (4.6h)$$

$$\partial_{\eta\Xi}\mathcal{U} \equiv \partial_{\Xi\eta}\mathcal{U} = \frac{-3K_\theta\alpha_p\theta}{\rho_{3D}C_p - K_\theta\alpha_p^2\theta} \quad (4.6i)$$

These materials constants are constrained by thermodynamics in that

$$0 < E_\theta < \frac{\rho_{1D}C_f}{\alpha_f^2\theta} \quad 0 < M_\theta < \frac{\rho_{2D}C_t}{\alpha_t^2\theta} \quad 0 < K_\theta < \frac{\rho_{3D}C_p}{\alpha_p^2\theta} \quad (4.7)$$

which ensure that their respective thermodynamic Jacobians cannot become singular. Singularities can and do occur, e.g., during a phase change in a crystal [46, 58], but such processes are not expected in our application.

4.1.3. Thermoelastic Models Suitable for Modeling Alveoli Subjected to Uniform Motions

We now write down our constitutive formulæ for quantifying uniform responses in thermoelastic solids of 1, 2 or 3 dimensions. They are the thermoelastic constitutive equations (4.4) with Helmholtz variables expressed in terms of the material constants defined in Eqn. (4.5) assigned to the internal energy density \mathcal{U} according to Eqn. (4.6), with outcomes of

$$\text{For 1D Materials:} \quad \begin{Bmatrix} d\eta \\ dF \end{Bmatrix} = \begin{bmatrix} C_f/\theta - E_\theta\alpha_f^2/\rho_{1D} & E_\theta\alpha_f/\rho_{1D} \\ -E_\theta\alpha_f & E_\theta \end{bmatrix} \begin{Bmatrix} d\theta \\ de \end{Bmatrix} \quad (4.8a)$$

$$\text{For 2D Materials:} \quad \begin{Bmatrix} d\eta \\ 2 dT \end{Bmatrix} = \begin{bmatrix} C_t/\theta - M_\theta\alpha_t^2/\rho_{2D} & 2M_\theta\alpha_t/\rho_{2D} \\ -2M_\theta\alpha_t & 4M_\theta \end{bmatrix} \begin{Bmatrix} d\theta \\ d\xi \end{Bmatrix} \quad (4.8b)$$

$$\text{For 3D Materials:} \quad \begin{Bmatrix} d\eta \\ -3 dP \end{Bmatrix} = \begin{bmatrix} C_p/\theta - K_\theta\alpha_p^2/\rho_{3D} & 3K_\theta\alpha_p/\rho_{3D} \\ -3K_\theta\alpha_p & 9K_\theta \end{bmatrix} \begin{Bmatrix} d\theta \\ d\Xi \end{Bmatrix} \quad (4.8c)$$

These constitutive formulæ, derived from the First and Second Laws of Thermodynamics, describe thermoelastic materials undergoing uniform motions through adiabatic processes.

The whole upper-left component in each matrix of Eqn. (4.8) represents a specific heat evaluated at constant strain, divided by temperature—a material property not easily measured. Whereas, the specific heat evaluated at constant pressure, viz., C_p with $C_t \equiv C_p$ and

$C_f \equiv C_p$, is more amenable to experiments, and is the property that one typically finds in published data tables.

There are four material constants for each dimension (e.g., for 1D materials they are ρ_{1D} , C_f , α_f and E_θ) with the latter three being defined according to Eqn. (4.5). The differential equations in Eqn. (4.8) have mathematical forms of hypo-elastic materials [59], which are preferred for incorporating constitutive equations into finite element packages.

4.2. Green Thermoelastic Membranes Subjected to Non-Uniform Motions

For modeling alveolar geometry via a dodecahedron, only the septal membranes are considered as being capable of supporting shears, and therefore, only they need to have their constitutive structure extended to include the non-uniform motions of shear, of which pure shears (squeeze) are likely to dominate over simple shears.

The First and Second Laws of Thermodynamics governing a reversible adiabatic process are described by the formula $d\mathcal{U} = \theta d\eta + \frac{1}{\rho} dW$, where dW is the mechanical power expended by stressing a body of mass density ρ . For the case of a 2D planar membrane, mass density $\rho \Leftarrow \rho_{2D}$ applies, with its change in mechanical work being expressed as [44, 45]

$$dW = \text{tr} \left(\begin{bmatrix} \mathcal{S}_{11} & \mathcal{S}_{12} \\ \mathcal{S}_{21} & \mathcal{S}_{22} \end{bmatrix} \begin{bmatrix} da/a & a dg/b \\ 0 & db/b \end{bmatrix} \right) = \pi d\xi + \sigma d\varepsilon + \tau d\gamma \quad (4.9a)$$

wherein the \mathcal{S}_{ij} are coefficients of Cauchy stress mapped into the co-ordinate frame of a membrane. The First and Second Laws describe an exact differential equation of the form

$$d\mathcal{U} = \theta d\eta + \frac{1}{\rho_{2D}} (\pi d\xi + \sigma d\varepsilon + \tau d\gamma) \quad (4.9b)$$

where $\{\pi, \sigma, \tau\}$ describes a set of intensive scalar-valued stresses whose thermodynamic conjugates $\{\xi, \varepsilon, \gamma\}$ describe a set of extensive scalar-valued strains. (This contrasts with the classic approach where the work done is decomposed into a scalar-valued isotropic part and a tensor-valued deviatoric part.) Pair (ξ, π) describes a dilation ξ caused by a surface tension π ; it is the uniform contribution to stress power discussed in §4.1 where $2 d\xi \Leftarrow A^{-1} dA$ and $\frac{1}{2}\pi \Leftarrow T$. Pair (ε, σ) describes a pure shear ε or squeeze caused by a normal-stress difference σ . And pair (γ, τ) describes an in-plane shear γ caused by a shear stress τ . Collectively, pairs (ε, σ) and (γ, τ) account for any non-uniform contributions to stress power, i.e., contributions from other than uniform dilations. These pairs are quantified in §4.2.3.

4.2.1. Constitutive Equations

Because a change in the internal energy $d\mathcal{U}$ governing a reversible adiabatic process is described by an exact differential [56], with $\mathcal{U}(\eta, \xi, \varepsilon, \gamma)$ in the case of a planar membrane, it follows that the constitutive response of a Green thermoelastic membrane is described by

$$\theta = \partial_\eta \mathcal{U}(\eta, \xi, \varepsilon, \gamma), \quad \pi = \rho \partial_\xi \mathcal{U}(\eta, \xi, \varepsilon, \gamma), \quad \sigma = \rho \partial_\varepsilon \mathcal{U}(\eta, \xi, \varepsilon, \gamma), \quad \tau = \rho \partial_\gamma \mathcal{U}(\eta, \xi, \varepsilon, \gamma) \quad (4.10a)$$

with strain symmetries requiring that

$$\mathcal{U}(\eta, \xi, \varepsilon, \gamma) = \mathcal{U}(\eta, \xi, -\varepsilon, \gamma) = \mathcal{U}(\eta, \xi, \varepsilon, -\gamma) = \mathcal{U}(\eta, \xi, -\varepsilon, -\gamma). \quad (4.10b)$$

Considering that each intensive variable, viz., θ , π , σ and τ , is actually a C^1 function of the set of extensive variables $(\eta, \xi, \varepsilon, \gamma)$, then the constitutive expressions in Eqn. (4.10a) can be recast into the following system of differential equations

$$\begin{Bmatrix} d\theta \\ d\pi \\ d\sigma \\ d\tau \end{Bmatrix} = \begin{bmatrix} \partial_{\eta\eta}\mathcal{U} & \partial_{\eta\xi}\mathcal{U} & \partial_{\eta\varepsilon}\mathcal{U} & \partial_{\eta\gamma}\mathcal{U} \\ \rho\partial_{\xi\eta}\mathcal{U} & \rho\partial_{\xi\xi}\mathcal{U} & \rho\partial_{\xi\varepsilon}\mathcal{U} & \rho\partial_{\xi\gamma}\mathcal{U} \\ \rho\partial_{\varepsilon\eta}\mathcal{U} & \rho\partial_{\varepsilon\xi}\mathcal{U} & \rho\partial_{\varepsilon\varepsilon}\mathcal{U} & \rho\partial_{\varepsilon\gamma}\mathcal{U} \\ \rho\partial_{\gamma\eta}\mathcal{U} & \rho\partial_{\gamma\xi}\mathcal{U} & \rho\partial_{\gamma\varepsilon}\mathcal{U} & \rho\partial_{\gamma\gamma}\mathcal{U} \end{bmatrix} \begin{Bmatrix} d\eta \\ d\xi \\ d\varepsilon \\ d\gamma \end{Bmatrix} \quad (4.11)$$

whose upper-left 2×2 sub-matrix also appears in Eqn. (4.3b), which governs the uniform contribution of a response. The above 4×4 matrix describes the full non-uniform response permissible by a Green thermoelastic membrane undergoing an adiabatic process.

For our application, it is reasonable to assume that the presence of any non-uniform motion will not cause an uniform response. Said differently, it is reasonable to assume that pure ε and simple γ shears will not effect a change in either temperature θ or surface tension π . Consequently, $\partial_{\eta\varepsilon}\mathcal{U} = \partial_{\eta\gamma}\mathcal{U} = \partial_{\xi\varepsilon}\mathcal{U} = \partial_{\xi\gamma}\mathcal{U} = 0$, and Eqn. (4.11) simplifies to

$$\begin{Bmatrix} d\theta \\ d\pi \\ d\sigma \\ d\tau \end{Bmatrix} = \begin{bmatrix} \partial_{\eta\eta}\mathcal{U} & \partial_{\eta\xi}\mathcal{U} & 0 & 0 \\ \rho\partial_{\xi\eta}\mathcal{U} & \rho\partial_{\xi\xi}\mathcal{U} & 0 & 0 \\ 0 & 0 & \rho\partial_{\varepsilon\varepsilon}\mathcal{U} & \rho\partial_{\varepsilon\gamma}\mathcal{U} \\ 0 & 0 & \rho\partial_{\gamma\varepsilon}\mathcal{U} & \rho\partial_{\gamma\gamma}\mathcal{U} \end{bmatrix} \begin{Bmatrix} d\eta \\ d\xi \\ d\varepsilon \\ d\gamma \end{Bmatrix}$$

with $\partial_{\varepsilon\eta}\mathcal{U} = \partial_{\gamma\eta}\mathcal{U} = \partial_{\varepsilon\xi}\mathcal{U} = \partial_{\gamma\xi}\mathcal{U} = 0$ because of Maxwell's relations. Converting the above internal energy formulation into its Helmholtz equivalent produces

$$\begin{Bmatrix} d\eta \\ d\pi \\ d\sigma \\ d\tau \end{Bmatrix} = \begin{bmatrix} 1/\partial_{\eta\eta}\mathcal{U} & -\partial_{\eta\xi}\mathcal{U}/\partial_{\eta\eta}\mathcal{U} & 0 & 0 \\ \rho\partial_{\xi\eta}\mathcal{U}/\partial_{\eta\eta}\mathcal{U} & \rho(\partial_{\xi\xi}\mathcal{U} - \partial_{\xi\eta}\mathcal{U} \cdot \partial_{\eta\xi}\mathcal{U}/\partial_{\eta\eta}\mathcal{U}) & 0 & 0 \\ 0 & 0 & \rho\partial_{\varepsilon\varepsilon}\mathcal{U} & \rho\partial_{\varepsilon\gamma}\mathcal{U} \\ 0 & 0 & \rho\partial_{\gamma\varepsilon}\mathcal{U} & \rho\partial_{\gamma\gamma}\mathcal{U} \end{bmatrix} \begin{Bmatrix} d\theta \\ d\xi \\ d\varepsilon \\ d\gamma \end{Bmatrix} \quad (4.12)$$

which provides the general structure for a Green thermoelastic membrane appropriate for our application. The uniform response (the upper-left 2×2 sub-matrix) and the non-uniform response (the lower-right 2×2 sub-matrix) are, by conjecture, decoupled in this constitutive construction.

4.2.2. Material Constants

The material model put forward here for a thermoelastic membrane has six material constants: a mass density ρ , a specific heat at constant surface tension C_π , a coefficient for areal thermal expansion at constant surface tension α_π , an areal modulus at constant temperature M_θ (2D version of a bulk modulus), a squeeze modulus at constant shear N_γ , and a shear modulus at constant squeeze G_ε . The specific heat C_π is defined as

$$C_\pi := \theta \partial_\theta \eta|_{s_{11}+s_{22}} = \theta \partial_\theta \eta|_\pi \quad (4.13a)$$

where θ is temperature, η is entropy, and $\pi = \mathcal{S}_{11} + \mathcal{S}_{22} = 2T$ is surface tension. $C_\pi = C_t \equiv C_p$ is commonly referred to in the literature as the specific heat at constant pressure. The areal coefficient for thermal expansion $\alpha_\pi = \alpha_t$ is defined as

$$\alpha_\pi := A^{-1} \partial_\theta A|_{\mathcal{S}_{11} + \mathcal{S}_{22}} = 2 \partial_\theta \xi|_\pi \quad (4.13b)$$

where $A = ab$ is area and $\xi = \ln \sqrt{A/A_0}$ is dilation. The areal modulus M_θ is defined as

$$M_\theta := A \partial_A (\mathcal{S}_{11} + \mathcal{S}_{22})|_\theta = \frac{1}{2} \partial_\xi \pi|_\theta. \quad (4.13c)$$

The in-plane squeeze modulus N_γ is defined as

$$N_\gamma := \Gamma \partial_\Gamma (\mathcal{S}_{11} - \mathcal{S}_{22})|_g = \frac{1}{2} \partial_\varepsilon \pi|_\gamma \quad (4.13d)$$

where $\sigma = \mathcal{S}_{11} - \mathcal{S}_{22}$ is a normal-stress difference, and $\Gamma = a/b$ is the stretch of squeeze with $\varepsilon = \ln \sqrt{\Gamma/\Gamma_0}$ being the strain of squeeze. And the in-plane shear modulus G_ε is defined as

$$G_\varepsilon := \Gamma \partial_g \mathcal{S}_{21}|_\Gamma = \partial_\gamma \tau|_\varepsilon \quad (4.13e)$$

where $\tau = \Gamma \mathcal{S}_{21}$ determines the shear stress with $\gamma = g - g_0$ being the shear strain.

Because $\ln(L/L_0) = \ln \sqrt{A/A_0}$ for uniform dilation (see Part 3), it necessarily follows then that if an axial coefficient for thermal expansion α_{1D} is available, viz., $\alpha_{1D} = L^{-1} \partial_\theta L|_F$, then set $\alpha_\pi = 2\alpha_{1D}$. Furthermore, because $\ln \sqrt[3]{V/V_0} = \ln \sqrt{A/A_0}$ under uniform dilatation, it also follows that if a volumetric coefficient for thermal expansion α_{3D} is available, viz., $\alpha_{3D} = V^{-1} \partial_\theta V|_P$, then set $\alpha_\pi = \frac{2}{3}\alpha_{3D}$.

In terms of these material constants, the thermoelastic membrane model of Eqn. (4.12) takes on the form of

$$\begin{Bmatrix} d\eta \\ d\pi \\ d\sigma \\ d\tau \end{Bmatrix} = \begin{bmatrix} C_\pi/\theta - M_\theta \alpha_\pi^2 / \rho_{2D} & 2M_\theta \alpha_\pi / \rho_{2D} & 0 & 0 \\ -2M_\theta \alpha_\pi & 4M_\theta & 0 & 0 \\ 0 & 0 & 2N_\gamma & 2\tau \\ 0 & 0 & 2\tau & G_\varepsilon \end{bmatrix} \begin{Bmatrix} d\theta \\ d\xi \\ d\varepsilon \\ d\gamma \end{Bmatrix} \quad (4.14)$$

where the upper-left 2×2 matrix comes from Eqn. (4.8b), which describes the uniform contribution to an overall planar response, while the lower-right 2×2 matrix comes from the following discussion: Given that $\Gamma := a/b$ denotes the stretch of squeeze with $\varepsilon := \ln \sqrt{\Gamma/\Gamma_0}$ defining the strain of squeeze, it follows that $d\Gamma = 2\Gamma d\varepsilon$. Squeeze ε , like dilation ξ , is logarithmic. By definition, $\tau := \Gamma \mathcal{S}_{21}$ whose differential change is $d\tau = \mathcal{S}_{21} d\Gamma + \Gamma d\mathcal{S}_{21}$ that, upon defining a shear modulus as $G_\varepsilon := \Gamma \partial_g \mathcal{S}_{21}|_\Gamma$, becomes $d\tau = 2\tau d\varepsilon + G_\varepsilon d\gamma$ because $dg = d\gamma$, with $d\tau = 2\tau d\varepsilon + G_\varepsilon d\gamma$ being the bottom formula in Eqn. (4.14). Maxwell's relation $\partial_{\gamma\varepsilon} \mathcal{U} = \partial_{\varepsilon\gamma} \mathcal{U}$ assigns 2τ to $\partial_{\varepsilon\gamma} \mathcal{U}$, while $N_\gamma := \Gamma \partial_\Gamma (\mathcal{S}_{11} - \mathcal{S}_{22})|_g$ establishes the squeeze modulus.

4.2.2.1. The Poisson Effect

The areal modulus M is ideally determined from an equibiaxial experiment. Assuming knowledge of its value, then given the following definition for Poisson's ratio

$$\nu := -\frac{db/b}{da/a}$$

it follows that the squeeze modulus N can be determined from an uniaxial experiment where traction is applied along that axis from which elongation a is measured; specifically,

$$N = \frac{1-\nu}{1+\nu}M \quad \text{provided that} \quad \mathcal{S}_{11} \neq 0 \quad \text{and} \quad \mathcal{S}_{21} = \mathcal{S}_{22} = 0.$$

Consequently, $\frac{1}{3}M \leq N \leq M$ given that $0 \leq \nu \leq 1/2$, so the squeeze modulus N plays an analogous role to the shear modulus μ of classical elasticity.

The conjugate pair approach presented here allows for a distinct shear modulus G that can take on any positive value. This is important because shear experiments done on soft tissues, which are few in number, tend to produce moduli that are many orders in magnitude smaller than their effective bulk modulus, e.g., in parenchyma their ratio is $K/G \approx 10^4$ (150 MPa vs. 10–54 kPa [60]). Classically, such a result is used to argue that the material can be modeled, to good approximation, as being incompressible—a 3D notion.

4.2.3. Conjugate Pairs \iff Tensor Components

The above planar strains and their rates are defined accordingly [44]

$$\xi := \ln \sqrt{\frac{a}{a_0} \frac{b}{b_0}} = \ln \sqrt{\frac{A}{A_0}} \quad \varepsilon := \ln \sqrt{\frac{b_0}{a_0} \frac{a}{b}} = \ln \sqrt{\frac{\Gamma}{\Gamma_0}} \quad \gamma := g - g_0 \quad (4.15a)$$

$$d\xi = \frac{1}{2} \left(\frac{da}{a} + \frac{db}{b} \right) = \frac{1}{2} \frac{dA}{A} \quad d\varepsilon = \frac{1}{2} \left(\frac{da}{a} - \frac{db}{b} \right) = \frac{1}{2} \frac{d\Gamma}{\Gamma} \quad d\gamma = dg \quad (4.15b)$$

where a, b, g are the physical attributes illustrated in Fig. 3.2, with a_0, b_0, g_0 being their reference values selected so that $\xi_0 = \varepsilon_0 = \gamma_0 = 0$, as established in Part 3 from a Gram-Schmidt factorization of the deformation gradient. These strains are two-state fields, independent of path—a tacit requirement of thermodynamics [56]. We point out that $\xi = \ln \sqrt{A/A_0}$ where $A = ab$ denotes area, and $\varepsilon = \ln \sqrt{\Gamma/\Gamma_0}$ where $\Gamma = a/b$ denotes the stretch of squeeze, i.e., the stretch of a pure shear deformation is in the sense of Becker [61] and Treloar [53]. Thermodynamic strains ξ, ε, γ and their differential rates $d\xi, d\varepsilon, d\gamma$ are exported by class `pentagon`, cf. Appendix F.

The conjugate stresses to the above planar strains are determined to be

$$\pi := \mathcal{S}_{11} + \mathcal{S}_{22} \quad \sigma := \mathcal{S}_{11} - \mathcal{S}_{22} \quad \tau := \Gamma \mathcal{S}_{21} \quad (4.15c)$$

which follow from Eqns. (4.9a & 4.15b) or, conversely, the physical components of surface tension \mathcal{S}_{ij} are established through

$$\mathcal{S}_{11} = \frac{1}{2}(\pi + \sigma) \quad \mathcal{S}_{22} = \frac{1}{2}(\pi - \sigma) \quad \mathcal{S}_{12} = \mathcal{S}_{21} = \Gamma^{-1}\tau \quad (4.15d)$$

when described in terms of their thermodynamic attributes π, σ and τ , plus the squeeze stretch Γ that must appear in τ so that $d\gamma$ associates with an exact differential—a thermodynamic requirement. The coupling appearing in the definition of shear stress, viz., $\Gamma\mathcal{S}_{21}$, is the reason why a coupling exists between squeeze and shear in constitutive equation (4.14).

The stress \mathcal{S} occurring in a septal plane, when evaluated in a pentagonal co-ordinate frame $(\vec{e}_1, \vec{e}_2, \vec{e}_3)$, only has non-zero components in the plane, i.e., in the upper-left 2×2 sub-matrix. These components map into a Cauchy stress σ expressed in the dodecahedral frame of reference $(\vec{i}, \vec{j}, \vec{k})$ via a component transformation of $\sigma_{ij} = P_{ik}Q_{kl}\mathcal{R}_{lm}\mathcal{S}_{mn}\mathcal{R}_{on}Q_{po}P_{jp}$. Because $\sigma_{12} = \sigma_{21}$, and because \mathbf{P} , \mathbf{Q} and \mathbf{R} are each orthogonal tensors, it necessarily follows that $\mathcal{S}_{12} = \mathcal{S}_{21}$. Matrix \mathbf{P} is a rotation of the pentagonal frame of reference about that of the dodecahedron's axes, matrix \mathbf{Q} is a rotation for re-indexing pentagonal co-ordinates, and matrix \mathbf{R} is a Gram-Schmidt rotation.

4.2.4. Continuity in Squeeze and Shear Responses in the Presence of a Re-Indexing

Whenever the pentagonal co-ordinate axes are re-indexed, in the sense that \mathbf{Q}_1 now describes the orthogonal re-indexing matrix \mathbf{Q} instead of \mathbf{Q}_0 in Eq. (3.21), then continuity of fields across such a co-ordinate relabeling requires squeeze and its rate to be redefined as

$$\varepsilon = \ln \sqrt{\frac{a_0}{b_0} \frac{b}{a}} = \ln \sqrt{\frac{\Gamma_0}{\Gamma}} \quad \text{with} \quad d\varepsilon = \frac{1}{2} \left(\frac{db}{b} - \frac{da}{a} \right) \quad (4.16)$$

while the conjugate stresses for squeeze and shear get redefined as

$$\sigma = \mathcal{S}_{22} - \mathcal{S}_{11} \quad \text{and} \quad \tau = \Gamma^{-1}\mathcal{S}_{21} \quad (4.17)$$

so that the work done by $\sigma d\varepsilon$ and $\tau d\gamma$ form contributions to the stress power that remain continuous across this co-ordinate switch. Components \mathcal{S}_{11} , \mathcal{S}_{21} and \mathcal{S}_{22} in Eq. (4.17) come from rotating the Cauchy stress σ according to the transformation $\mathcal{S} = \mathbf{R}^T \mathbf{Q}^T \mathbf{P}^T \sigma \mathbf{P} \mathbf{Q} \mathbf{R}$.

This process is necessary because, although the conjugate pair (ξ, π) is invariant of frame, conjugate pairs (ε, σ) and (γ, τ) are not.

4.3. Modeling an Alveolus

Only one-third of the cross-sectional area of an alveolar chord, and only one-half of the wall thickness of an alveolar septum associate with any given dodecahedron [26]. Specifically, a third of the total force carried by a septal fiber belongs with a given alveolus, with the remaining two-thirds of the transmitted force belonging to its two adjoining alveoli. Likewise, only half of the surface traction carried along a septal membrane belongs with a given alveolus, with the other half of its surface traction belonging to its adjacent alveolus. Like statements apply for their entropies.

About 75% of the acting transpulmonary pressure (the difference between pleural and alveolar pressures) is carried by the alveolar structure with the remaining 25% being carried by the pleural membrane encasing the lung [62].

transpulmonary pressure	4 cm H ₂ O		
Age	15–35	36–45	> 65
collagen: \sqrt{D} , $(\mu\text{m})^{1/2}$	0.952 ± 0.242	0.958 ± 0.255	1.045 ± 0.270
elastin: \sqrt{D} , $(\mu\text{m})^{1/2}$	0.957 ± 0.239	0.970 ± 0.213	1.093 ± 0.274
transpulmonary pressure	14 cm H ₂ O		
Age	15–35	36–45	> 65
collagen: \sqrt{D} , $(\mu\text{m})^{1/2}$	0.955 ± 0.246	0.994 ± 0.237	1.054 ± 0.279
elastin: \sqrt{D} , $(\mu\text{m})^{1/2}$	0.956 ± 0.237	0.988 ± 0.263	1.079 ± 0.281

Table 4.1: Mean and standard deviations in variance for the square root of septal chord diameters \sqrt{D} reported by Sobin *et al.* [34]. These septal chords are comprised of collagen and elastin fibers, which act largely independently, and therefore are considered to be loaded in parallel with one another.

4.3.1. Constraints/Assumptions for Alveoli Subjected to Shock Waves

Because the purpose for the alveolar model being constructed is to better understand alveolar behavior as a shock wave passes over it, there are certain assumptions that we impose on our model that under normal or different physiologic conditions might otherwise not apply.

First: An alveolus is considered to be an adiabatic pressure vessel in which air and heat cannot move into or out of as a shock wave passes over it, simply because the wave speed is too fast. There is insufficient time for these transports to occur.

Second: Temperature remains constant across a shock wave front traveling through a compressible gas [63]. It is therefore assumed that alveolar temperature remains at body temperature whenever a lung is subjected to a shock wave. All changes in alveolar entropy are therefore caused by changes in alveolar strain.

Third: The air/membrane interface of an alveolus is lined with a surfactant, which is a thin bi-lipid film that plays a substantial role in normal lung function. This film reduces alveolar surface tension to avert total lung collapse at maximum exhale [64]. Even so, some alveoli still collapse, getting re-recruited during a later breath. Models have been proposed for both surfactant [65] and alveolar recruitment [66], but these effects are not included here as they are not thought to play a significant role in lung mechanics in the presence of a shock wave.

4.3.2. Modeling a Septal Chord Subjected to a Shock Wave

Alveoli are biologic structures constructed of septal chords that circumscribe alveolar membranes that envelope an alveolar sac whereat gas exchange occurs. These chords are comprised of individual collagen and elastin fibers loaded in parallel [34, 67]. The extent of elastic energy stored within a chord will depend upon the diameters D^c and D^e and length L of these individual fibers.⁶ Let superscript ‘*c*’ denote collagen, and superscript ‘*e*’ denote

⁶Sobin *et al.* [34] considered that the stored energy of chords also depends upon their curvature, which they measured and quantified, i.e., they considered these chords to be beams. However, with a slenderness

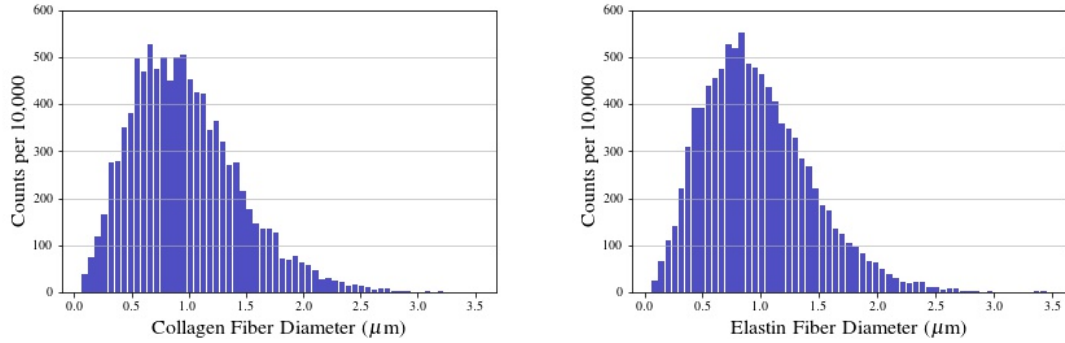


Figure 4.1: Typical histograms for collagen and elastin chord diameters pertaining to the statistics reported in Table 4.1. Their tails weigh heavy at the larger diameters, because their distributions are normal in the square roots of their diameters. These histograms are identical for all practical purposes.

elastin. Sobin *et al.* [34] determined that the square root of their diameters \sqrt{D} distribute normally, with a mean $\bar{D}^{1/2}$ and standard deviation $\sigma_{\sqrt{D}}$ that also depend upon age and transpulmonary pressure, as presented in Table 4.1 and illustrated in Fig. 4.1.

The collagen and elastin fibers that make up a septal chord have the same length, they experience the same strain, and they exist at the same temperature; therefore, we employ Eqn. (4.8a) as the governing constitutive equation to describe their mechanical behavior

$$\begin{Bmatrix} d\eta \\ dF \end{Bmatrix} = \begin{bmatrix} C/\theta - E\alpha^2/\rho & E\alpha/\rho \\ -E\alpha & E \end{bmatrix} \begin{Bmatrix} d\theta \\ de \end{Bmatrix}$$

wherein C is a specific heat at constant pressure, α is a linear coefficient of thermal expansion, E is an elastic modulus with units of force, and ρ is its mass density per unit length. Because $d\theta = 0$ across a wave front [63], this general system reduces to

$$dF = E de \quad \text{and} \quad \rho d\eta = E\alpha de = \alpha dF$$

wherein the first equation establishes a change in force carried by a chord, while the second equation establishes a change in entropy per unit length of chord. Again, we emphasize that material constant ρ has units of mass per unit length of line, while material constant E has units of force in the above formulæ. Below we change the units of these material constants to better address our application.

4.3.2.1. Chordal Forces and Entropies

The constitutive equations that govern alveolar chords, suitable for studying shock waves in lungs, are described by the following system of differential equations

$$F = F^c + F^e \quad dF^c = A_0^c e^{-e} E^c(e, F^c) de \quad dF^e = A_0^e e^{-e} E^e(e, F^e) de \quad (4.18a)$$

$$S = S^c + S^e \quad dS^c = \alpha^c dF^c \quad dS^e = \alpha^e dF^e \quad (4.18b)$$

ratio of $\bar{L}/\bar{D} = 102 \pm 12$, which we obtained from their data, it is reasonable to model them as rods, not beams. Consequently, the dodecahedral truss to be considered is a pinned truss, not a rigid truss, thereby greatly simplifying the boundary value problem.

where F is the total force carried by a septal chord, with F^c and F^e being the forces carried by its collagen and elastin fiber constituents with cross-sectional areas of $A^c = A_0^c \exp(-e)$ and $A^e = A_0^e \exp(-e)$, while S is the chord's entropy with S^c and S^e being the entropies of its two parts. For initial conditions we use forces $F_0^c = F^c|_{L=L_0} = 0$ and $F_0^e = F^e|_{L=L_0} = 0$, and we use entropies $S_0^c = \rho^c V_0^c \eta_0^c$ and $S_0^e := \rho^e V_0^e \eta_0^e$, wherein η_0^c and η_0^e are the initial entropy densities per unit volume, with V_0^c and V_0^e being the volumes of these fibers in their traction-free state. It remains to establish the two tangent moduli $E^c(e, F^c)$ and $E^e(e, F^e)$ for collagen and elastin that, in these formulæ, now have units of stress in that $A^c E^c$ and $A^e E^e$ have units of force. These tangent moduli are functions of state, see e.g., Ref. [68].

Collagen is a fiber comprised of numerous, long, slender, wavy filaments whose waviness, known as crimp, straightens under sufficient deformation [69, 70]. Elastin is a linked fiber network, much like an elastomer, whose filaments between crosslinks rotate to align with an axis of loading under sufficient deformation [71, 72]. Consequently, collagen and elastin both recruit constituent filaments with increasing deformation into an overall, load-bearing, fiber response. The internal energies of collagen and elastin may therefore be thought of as being comprised of a configurational energy and a strain energy. As such, both collagen and elastin are modeled as Freed-Rajagopal biologic fibers, which are described in terms of two such energies. Their model is derived from the theory of implicit elasticity in Appendix A. According to their model, tangent compliances for collagen and elastin are described by

$$\frac{1}{E^c} = \frac{e_t^c - e_1^c}{e_T^c E_1^c + 2s^c} + \frac{1}{E_2^c} \quad e_1^c = e - \alpha^c(\theta - \theta_0) - \frac{s^c}{E_2^c} \quad s^c = \frac{F^c}{A_0^c e^{-e}} \quad (4.19a)$$

$$\frac{1}{E^e} = \frac{e_t^e - e_1^e}{e_T^e E_1^e + 2s^e} + \frac{1}{E_2^e} \quad e_1^e = e - \alpha^e(\theta - \theta_0) - \frac{s^e}{E_2^e} \quad s^e = \frac{F^e}{A_0^e e^{-e}} \quad (4.19b)$$

recalling that $\theta = \theta_0$ in our application, so the intermediate strains defined in the second column simplify. Material constants E_1^c and E_2^c are asymptotic moduli for collagen such that $E_1^c \leq E^c \leq E_2^c$, while E_1^e and E_2^e are asymptotic moduli for elastin such that $E_1^e \leq E^e \leq E_2^e$, all of which here have units of stress (barye = dyne/cm²), with e_t^c and e_t^e being their respective transition strains, cf. Appendix A. The material properties needed to model septal chords are listed in Tables 4.1 & 4.2. From Eqn. (4.7), these moduli are bound from above by $E_{\max}^c = 4.9 \times 10^9$ barye (dyne/cm²) and $E_{\max}^e = 1.7 \times 10^9$ barye. We therefore observe that E_2^c and E_2^e are about 100 times smaller than E_{\max}^c and E_{\max}^e .

The actual force and entropy carried by an individual septal chord in our alveolar model will be one third of their calculated values from Eqn. (4.18), because each chord is shared between three neighboring alveoli.

4.3.3. Modeling an Alveolar Membrane Subjected to a Shock Wave

Membranes have elastic moduli, whose dimensions are appropriate for two space, plus a bending stiffness excited by curvature. This bending energy is proportional to $(h/r)^2$, with h being membrane thickness (height) and r being its radius of curvature. Consequently, bending stresses can be neglected when compared with their in-plane stresses whenever $h \ll r$, which is the supposition here. This is in concert with our assumption that the septal

Collagen		
ρ^c [gr/cm ³]	1.34	Fels [73]
η_0^c [erg/gr.K]	3.7×10^4	
C^c [erg/gr.K]	1.7×10^7	Kanagy [74]
α^c [1/C]	1.8×10^{-4}	Weir [75]
e_t^c	0.09	estimated from TLC $\approx 30\%$
E_1^c [barye]	5.0×10^5	
E_2^c [barye]	3.0×10^7	
Elastin		
Parameter	Value	Reference
ρ^e [gr/cm ³]	1.31	Lillie & Gosline [76]
η_0^e [erg/gr.K]	3.4×10^4	Shadwick & Gosline [77]
C^e [erg/gr.K]	4.2×10^7	Kakivaya & Hoeve [78]
α^e [1/C]	3.2×10^{-4}	Lillie & Gosline [76]
e_t^e	0.4	Shadwick & Gosline [77]
E_1^e [barye]	2.3×10^6	Urry [72, Fig. 18]
E_2^e [barye]	1.0×10^7	Lillie & Gosline [79, Fig. 5]

Table 4.2: Physical properties for hydrated collagen and elastin fibers. Collagen denatures at around 60°C [80], i.e., above this temperature collagen will shrink rapidly—an effect not modeled here.

chords are modeled as rods, not beams, because of their slenderness ratio. Furthermore, these septa tend to be flat because there are roughly equal pressures acting on both sides of these membranes, eliminating bending effects and, we surmise, suppressing wrinkling, too.

From a mechanics perspective, we know a great deal more about alveolar chords than we know about alveolar septa. More judgment will therefore be required in our construction and parameterization of a material model for alveolar membranes.

A typical alveolar septum is 4-5 μm in width [55]. They are comprised of an outside layer of epithelial cells that encase capillaries made of endothelial cells along with a basement membrane that is composed of unorganized collagen and elastin filaments, plus proteoglycans and other extracellular proteins. This basement membrane, roughly at mid-plane in an alveolar septum, has a width of about 0.5 μm [13]. Inertial forces generated by these membranes are to be based upon a membrane thickness of $\sim 5 \mu\text{m}$ with an approximate density of water, while the structural forces that they carry are to be based upon a basement membrane thickness of $\sim 0.5 \mu\text{m}$.

It is not known how much of the mechanical load is actually carried by the cells in an alveolar septum vs. the extracellular basement membrane they encase, but it is generally thought that this basement membrane carries the majority of the load [55]. Therefore, by diminishing the moduli that are appropriate for describing a basement membrane with thickness $\sim 0.5 \mu\text{m}$ by a factor of 10, we get an estimate for the effective septal moduli appropriate when modeling a whole septal membrane with thickness $\sim 5 \mu\text{m}$. We use the model parameters obtained for a visceral pleura membrane [44] and the above conjecture to

model the alveolar septa.

From Eqn. (4.14), alveolar septa have a constitutive response that, in the presence of a shock wave, is governed by

$$\begin{Bmatrix} \rho d\eta \\ d\pi \\ d\sigma \\ d\tau \end{Bmatrix} = \begin{bmatrix} 2M\alpha & 0 & 0 \\ 4M & 0 & 0 \\ 0 & 2N & 2\tau \\ 0 & 2\tau & G \end{bmatrix} \begin{Bmatrix} d\xi \\ d\varepsilon \\ d\gamma \end{Bmatrix}$$

because $d\theta = 0$. Here α is the areal coefficient of thermal expansion, M is the dilation modulus, N is the squeeze modulus, G is the shear modulus (all moduli have units of force per length of line), and ρ is the mass density per unit area. Below we change the units of these moduli, viz., from force per unit length to force per unit area, i.e., stress, so that they are in a form more familiar to the user.

4.3.3.1. Surface Stresses and Entropy

The intensive thermodynamic variables π , σ and τ have units of force per unit length of line. We convert these into stresses by defining $s_\pi := \pi/h$, $s_\sigma := \sigma/h$ and $s_\tau := \tau/h$, which have units of force per unit area. Herein $h := h_0 e^{-2\xi}$ denotes membrane thickness established by assuming membrane volume is preserved, with h_0 being its reference value. Adopting these definitions allows the above system of equations to be rewritten as

$$\begin{Bmatrix} dS \\ ds_\pi \\ ds_\sigma \\ ds_\tau \end{Bmatrix} = \begin{bmatrix} 2\rho V_0 M\alpha & 0 & 0 \\ 4M + 2s_\pi & 0 & 0 \\ 2s_\sigma & 2N & 2s_\tau \\ 2s_\tau & 2s_\tau & G \end{bmatrix} \begin{Bmatrix} d\xi \\ d\varepsilon \\ d\gamma \end{Bmatrix} \quad (4.20)$$

wherein moduli M , N and G now all have units of stress. Here S is the entropy of a membrane with initial condition $S_0 = \rho V_0 \eta_0$ whose volume is V_0 and whose entropy density per unit mass is η_0 with ρ now describing mass density per unit volume.

Collagen and elastin appear as thin filaments randomly oriented and somewhat uniformly dispersed throughout a basement membrane, unlike the strongly aligned fibers that appear in septal chords. Consequently, for our purposes, we model this collective ensemble of tissue and structure types as a homogeneous isotropic membrane modeled after the Freed-Rajagopal biologic fiber [68] that we have extended to membranes in Appendix A, specifically

$$\frac{1}{M} = \frac{\xi_t - \xi_1}{M_1 \xi_t + s_\pi/2} + \frac{1}{M_2} \quad \xi_1 = \xi - \frac{1}{2}\alpha(\theta - \theta_0) - \frac{s_\pi}{4M_2} \quad s_\pi := \frac{\pi}{h_0 e^{-2\xi}} \quad (4.21a)$$

$$\frac{1}{N} = 2 \left(\frac{\varepsilon_t - |\varepsilon_1|}{N_1 \varepsilon_t + 2\gamma s_\tau} + \frac{1}{N_2} \right) \quad \varepsilon_1 = \varepsilon - \frac{s_\sigma - 2\gamma s_\tau}{N_2} \quad s_\sigma := \frac{\sigma}{h_0 e^{-2\xi}} \quad (4.21b)$$

$$\frac{1}{G} = \Gamma \left(\frac{\gamma_t - 2\varepsilon\gamma - |\gamma_1|}{G_1 \gamma_t + 2\varepsilon s_\tau} + \frac{1 - 2\varepsilon}{G_2} \right) \quad \gamma_1 = \gamma - \frac{(1 - 2\varepsilon)s_\tau}{G_2} \quad s_\tau := \frac{\tau}{h_0 e^{-2\xi}} \quad (4.21c)$$

while recalling that $\theta = \theta_0$, which thereby simplifies ξ_1 because temperature across a shock wave front is constant [63]. The compliant tangent moduli of M_1 , N_1 and G_1 and the stiff,

terminal, tangent moduli of M_2, N_2 and G_2 bound their responses in that $M_1 \leq M \leq M_2$, $N_1 \leq N \leq N_2$ and $G_1 \leq G \leq G_2$, with gradual transitions between their asymptotic bounds being centered around strains of ξ_t, ε_t and γ_t . Here $h_0 = 4.5 \pm 0.5 \mu\text{m}$ is the initial thickness of a membrane with $h = h_0 \exp(-2\xi)$ establishing its current thickness, obtained via an assumption of material incompressibility, remembering that $\xi = \ln \sqrt{A/A_0}$.

Matsuda *et al.* [67] found the diameters of collagen and elastin fibers that circumscribe an alveolar mouth to be about 5-7 times larger than those of their septal chords. The alveolar mouth, with its thicker fibers and open face that attach an alveolus to an alveolar duct, is modeled here as a phantom face, viz., with fibers sized like any of the other eleven pentagonal elements comprising a dodecahedron, and a twelfth face [11]. Kimmel & Budiansky addressed this point via a private communication they had with Prof. T. A. Wilson. They wrote [28]:

“Professor T. A. Wilson notes that the present model does not take explicit account of either alveolar openings or their fibrous boundaries. Wilson suggests that the elastic resistance of the ring boundaries tends to make up for the missing surface tension in the holes, so that neglect of both effects may be self-compensating.”

This conjecture of Kimmel & Budiansky, along with the experimental findings of Matsuda *et al.*, will enable us to estimate the surface traction carried by a single alveolar membrane, relative to the tractions carried by its alveolar chords. In other words, this provides an avenue for parameterizing the membrane model in an otherwise void of relevant experimental data needed to estimate these parameters.

To convert these membrane stresses into nodal forces requires finite element technology, which is discussed in Part 6. We recall that only half of their nodal tractions will apply to the alveolus being modeled, because each septum is shared between two alveoli.

4.3.4. Modeling an Alveolar Volume Subjected to a Shock Wave

Alveoli are connected to bronchial trees via alveolar ducts. Under normal conditions, air moves in and out of the alveoli via these ducts. However, when subjected to a stress wave passing over an alveolus, there is no time for the transport of air to take place. Hence, we can consider the air (and heat) within an alveolus to become ‘trapped’, and the pressure to be uniform therein. The governing process is therefore adiabatic. It is under this condition that we model the volumetric response of alveolar sacs.

If one considers the saturated air/fluid within an alveolus to be an ideal gas, then [81]

$$PV = nR\theta \quad \text{or} \quad \frac{PV}{\theta} = \frac{P_0 V_0}{\theta_0} = nR = \text{constant} \quad (4.22)$$

where, in our case, P_0 is taken to be atmospheric pressure at sea level (1 bar or 10^5 Pa or 10^6 barye), with V_0 being that alveolar volume whereat alveolar pressure and plural pressure are both atmospheric, while $\theta_0 = 37^\circ\text{C} = 310$ K is assigned as body temperature. Parameter n is the molar content of gas within an alveolus, with R being the universal gas constant.

The material constants associated with an ideal gas contained within an adiabatic enclosure are

$$\alpha := \frac{1}{V} \left. \frac{\partial V}{\partial \theta} \right|_P = \frac{1}{\theta_0} \frac{P_0 V_0}{PV} \quad \text{and} \quad K := -\frac{1}{V} \left. \frac{\partial P}{\partial V} \right|_\theta = P_0 \frac{V_0 \theta}{\theta_0 V} \quad (4.23)$$

with the other two material constants pertaining to moist air at body temperature⁷ being its mass density ρ of 1.125×10^{-3} gr/cm³ and its specific heat C at constant pressure of 1.007×10^7 erg/gr.K, constrained by the fact that $K < K_{\max} = \rho C / \alpha^2 \theta \approx \rho C \theta$.

An alveolar sac is modeled as an adiabatic pressure vessel filled with an ideal gas whose Helmholtz formulation (4.8c) is described by

$$\begin{Bmatrix} \rho d\eta \\ -3 dP \end{Bmatrix} = \begin{bmatrix} \rho C / \theta - K \alpha^2 & 3K\alpha \\ -3K\alpha & 9K \end{bmatrix} \begin{Bmatrix} d\theta \\ d\Xi \end{Bmatrix} \quad (4.24)$$

that, again, because $\theta = \theta_0$ across a wave front [63], simplifies the governing system of differential equations (4.24) to

$$dP = -3K d\Xi \quad \text{and} \quad \rho d\eta = 3K\alpha d\Xi = -\alpha dP$$

where we recall that $d\Xi = \frac{1}{3} V^{-1} dV$. Like before, we do not want the entropy per unit mass or per unit volume; rather, we prefer to know what the entropy of its structural constituent is, in this case, the gas within an alveolar sac. To find this, we multiply the right-hand expression through by volume V_0 , after which we integrate both differential equations to get

$$\frac{P}{P_0} = \frac{V_0}{V} \quad \therefore \quad \alpha = \frac{1}{\theta_0} \quad \implies \quad S = S_0 + \frac{V_0(P_0 - P)}{\theta_0} \quad (4.25)$$

where S is the desired entropy of an alveolar volume with $S_0 = \rho\eta_0 V_0$ given that $\rho\eta_0$ is the entropy per unit volume of humid air at body temperature and atmospheric pressure, viz., $\rho\eta_0 = 7.770 \times 10^4$ erg/cm³.K. The above results are in agreement with the ideal gas law, as they should be, because α and K in Eqn. (4.23) were both derived from this law.

To convert pressure P into nodal forces in our alveolar model requires finite element technology, which is discussed in Part 6.

4.4. Code Verification and Constitutive Parameterization

⁷The physical properties listed for air were taken from the website www.peacesoftware.de hosted by Berndt Wischniewski.

Part 5

Numerical Integrators

This analysis tool that models alveoli using dodecahedra requires numerical methods for performing integrations. These numerical methods are discussed below.

5.1. ODE Solvers

The various constitutive equations that describe our alveolar model present themselves as ordinary differential equations that need to be integrated. To this end, we employ the PECE (Predict, Evaluate, Correct, re-Evaluate) algorithms of Freed [82] which are suitable for solving stiff systems of first- and second-order, ordinary, differential equations. These methods are based upon Gear's well-known second-order backward difference formula (BDF2).

Time t is considered to be the independent variable, discretized over an interval in time $[t_0, t_n]$ for which N solutions are to be extracted at nodes $n = 1, 2, \dots, N$ spaced at uniform intervals in time with a common step size of $h = (t_n - t_0)/N$ separating them. This is the *global step size* at which solutions are to be gathered. A dynamically controlled *local step size*, whose size is adjusted to manage truncation error, is implemented into the code according to a scheme put forward in Ref. [83]. Solver interfaces are listed in Appendix H.

5.1.1. PECE Solver for First-Order ODEs

Let \mathbf{x} be a vector of dependent variables obeying a differential equation of evolution $d\mathbf{x}(t)/dt = \dot{\mathbf{x}} = \mathbf{f}(t, \mathbf{x})$ subject to an initial condition $\mathbf{x}_0 = \mathbf{x}(t_0)$. One may think of \mathbf{x} as being displacements whose rates $\dot{\mathbf{x}} = \mathbf{v}$ are velocities. We find it useful to express $\dot{\mathbf{x}}$ as \mathbf{v} .

The method put forward here incrementally solves such an ODE, returning solutions associated with the next moment in time t_{n+1} , i.e., it acquires \mathbf{x}_{n+1} , given knowledge of the previous \mathbf{x}_{n-1} and current \mathbf{x}_n solutions plus their rates \mathbf{v}_{n-1} and \mathbf{v}_n with the corrector also depending upon \mathbf{v}_{n+1} , i.e., the corrector is an implicit method.

5.1.1.1. Start-Up Algorithm

Multi-step methods are not self starting; consequently, Heun's method (a forward-Euler predictor with a trapezoidal corrector) is used here to start our integrator; specifically,

$$\text{Predict} \quad \mathbf{x}_1^p = \mathbf{x}_0 + h\mathbf{v}_0 + \mathcal{O}(h^2) \quad (5.1a)$$

$$\text{Evaluate} \quad \mathbf{v}_1^p = \mathbf{f}(t_1, \mathbf{x}_1^p) \quad (5.1b)$$

$$\text{Correct} \quad \mathbf{x}_1 = \mathbf{x}_0 + \frac{1}{2}h(\mathbf{v}_1^p + \mathbf{v}_0) + \mathcal{O}(h^3) \quad (5.1c)$$

$$\text{Re-Evaluate} \quad \mathbf{v}_1 = \mathbf{f}(t_1, \mathbf{x}_1) \quad (5.1d)$$

wherein $\mathbf{v}_0 = \mathbf{f}(t_0, \mathbf{x}_0)$ and $t_1 = t_0 + h$. The correct/re-evaluate steps can be iterated over until a convergence criterion is satisfied, if need be.

5.1.1.2. Two-Step ODE Solver

The two-step method of Freed [82] for solving 1st order ODEs is

$$\text{Predict} \quad \mathbf{x}_{n+1}^p = \frac{1}{3}(4\mathbf{x}_n - \mathbf{x}_{n-1}) + \frac{2}{3}h(2\mathbf{v}_n - \mathbf{v}_{n-1}) + \mathcal{O}(h^3) \quad (5.2a)$$

$$\text{Evaluate} \quad \mathbf{v}_{n+1}^p = \mathbf{f}(t_{n+1}, \mathbf{x}_{n+1}^p) \quad (5.2b)$$

$$\text{Correct} \quad \mathbf{x}_{n+1} = \frac{1}{3}(4\mathbf{x}_n - \mathbf{x}_{n-1}) + \frac{2}{3}h\mathbf{v}_{n+1}^p + \mathcal{O}(h^3) \quad (5.2c)$$

$$\text{Re-Evaluate} \quad \mathbf{v}_{n+1} = \mathbf{f}(t_{n+1}, \mathbf{x}_{n+1}) \quad (5.2d)$$

where $t_{n+1} = t_n + h$. This corrector is the well-known BDF2 formula of Gear to which Freed provided a predictor. The correct/re-evaluate steps can be iterated over until a convergence criterion is satisfied, if need be. Implementation of this integrator is found in Appendix H.1.

For both the predictor and corrector, solution \mathbf{x} has a weight of 1, while its rate \mathbf{v} has a weight of $\frac{2}{3}h$; hence, this predictor/corrector pair is consistent.

5.1.2. PECE Solver for Second-Order ODEs

Let \mathbf{x} be a vector of dependent variables obeying a differential equation of evolution $d^2\mathbf{x}(t)/dt^2 = \ddot{\mathbf{x}} = \mathbf{f}(t, \mathbf{x}, \dot{\mathbf{x}})$ subject to initial conditions $\mathbf{x}_0 = \mathbf{x}(t_0)$ and $\dot{\mathbf{x}}_0 = \dot{\mathbf{x}}(t_0)$. One may think of \mathbf{x} as being displacements and their rates $\mathbf{v} = \dot{\mathbf{x}}$ as being velocities with $\mathbf{a} = \dot{\mathbf{v}} = \ddot{\mathbf{x}}$ representing accelerations. We find it useful to express $\dot{\mathbf{x}}$ as \mathbf{v} and $\ddot{\mathbf{x}}$ as \mathbf{a} .

The method put forward here incrementally solves such an ODE, returning solutions associated with the next moment in time t_{n+1} , i.e., it acquires \mathbf{x}_{n+1} & \mathbf{v}_{n+1} given knowledge of the previous \mathbf{x}_{n-1} & \mathbf{v}_{n-1} and current \mathbf{x}_n & \mathbf{v}_n solutions plus their accelerations \mathbf{a}_{n-1} and \mathbf{a}_n with the corrector also depending upon \mathbf{a}_{n+1} , i.e., the corrector is an implicit method.

5.1.2.1. Start-Up Algorithm

Multi-step methods are not self starting, so a one-step method is needed to take the first step of integration, specifically

$$\text{Predict} \quad \mathbf{x}_1^p = \mathbf{x}_0 + h\mathbf{v}_0 + \frac{1}{2}h^2\mathbf{a}_0 + \mathcal{O}(h^3) \quad (5.3a)$$

$$\mathbf{v}_1^p = \mathbf{v}_0 + h\mathbf{a}_0 + \mathcal{O}(h^2) \quad (5.3b)$$

$$\text{Evaluate} \quad \mathbf{a}_1^p = \mathbf{a}(t_1, \mathbf{x}_1^p, \mathbf{v}_1^p) \quad (5.3c)$$

$$\text{Correct} \quad \mathbf{x}_1 = \mathbf{x}_0 + \frac{1}{2}h(\mathbf{v}_1^p + \mathbf{v}_0) - \frac{1}{12}h^2(\mathbf{a}_1^p - \mathbf{a}_0) + \mathcal{O}(h^4) \quad (5.3d)$$

$$\mathbf{v}_1 = \mathbf{v}_0 + \frac{1}{2}h(\mathbf{a}_1^p + \mathbf{a}_0) + \mathcal{O}(h^3) \quad (5.3e)$$

$$\text{Re-Evaluate} \quad \mathbf{a}_1 = \mathbf{a}(t_1, \mathbf{x}_1, \mathbf{v}_1) \quad (5.3f)$$

wherein $\mathbf{a}_0 = \mathbf{f}(t_0, \mathbf{x}_0, \mathbf{v}_0)$ and $t_1 = t_0 + h$. The correct/re-evaluate steps can be iterated over until a convergence criterion is satisfied, if need be.

5.1.2.2. Two-Step ODE Solver

The two-step method of Freed [82] for solving 2nd order ODEs is

$$\begin{aligned} \text{Predict} \quad \mathbf{x}_{n+1}^p &= \frac{1}{3}(4\mathbf{x}_n - \mathbf{x}_{n-1}) + \frac{1}{6}h(3\mathbf{v}_n + \mathbf{v}_{n-1}) \\ &\quad + \frac{1}{36}h^2(31\mathbf{a}_n - \mathbf{a}_{n-1}) + \mathcal{O}(h^4) \end{aligned} \quad (5.4a)$$

$$\mathbf{v}_{n+1}^p = \frac{1}{3}(4\mathbf{v}_n - \mathbf{v}_{n-1}) + \frac{2}{3}h(2\mathbf{a}_n - \mathbf{a}_{n-1}) + \mathcal{O}(h^3) \quad (5.4b)$$

$$\text{Evaluate} \quad \mathbf{a}_{n+1}^p = \mathbf{a}(t_{n+1}, \mathbf{x}_{n+1}^p, \mathbf{v}_{n+1}^p) \quad (5.4c)$$

$$\begin{aligned} \text{Correct} \quad \mathbf{x}_{n+1} &= \frac{1}{3}(4\mathbf{x}_n - \mathbf{x}_{n-1}) + \frac{1}{24}h(\mathbf{v}_{n+1}^p + 14\mathbf{v}_n + \mathbf{v}_{n-1}) \\ &\quad + \frac{1}{72}h^2(10\mathbf{a}_{n+1}^p + 51\mathbf{a}_n - \mathbf{a}_{n-1}) + \mathcal{O}(h^4) \end{aligned} \quad (5.4d)$$

$$\mathbf{v}_{n+1} = \frac{1}{3}(4\mathbf{v}_n - \mathbf{v}_{n-1}) + \frac{2}{3}h\mathbf{a}_{n+1}^p + \mathcal{O}(h^3) \quad (5.4e)$$

$$\text{Re-Evaluate} \quad \mathbf{a}_{n+1} = \mathbf{a}(t_{n+1}, \mathbf{x}_{n+1}, \mathbf{v}_{n+1}) \quad (5.4f)$$

where $t_{n+1} = t_n + h$. An implementation of this integrator is described in Appendix H.2.

The above PECE solver for velocity \mathbf{v} is the same method presented in Eq. (5.2), so this predictor/corrector pair is consistent. Likewise, in both the predictor and corrector for displacement \mathbf{x} , contributions from the solution \mathbf{x} have a weight of 1, contributions from the velocities \mathbf{v} have a weight of $\frac{2}{3}h$, and contributions from the accelerations \mathbf{a} have a weight of $\frac{5}{6}h^2$; hence, this predictor/corrector pair is consistent, too.

5.2. Quadrature Rules for a Regular Pentagon

Three, Gauss, quadrature rules for a regular pentagon described in its natural coordinate system, i.e., oriented according to Fig. 2.2, are presented in Table 5.1. These quadratures exactly integrate polynomials of order 1, 3 and 5, respectively. They were supplied to the authors by Prof. N. Sukumar from the University of California at Davis, which he derived for us at our request using a methodology that he published in [84]. In that document, the authors derived formulæ for determining the nodes and weights for a class of generalized, Gaussian, quadrature rules, which they then applied to pentagons, hexagons, heptagons and octagons, of which they only published their nodes and weights of quadrature for the hexagon, as it tiles two space. The node for the 1st order method is located at the centroid of the pentagon. Nodes for the 3rd and 5th order methods of Table 5.1 are displayed in Fig. 5.1.

The Gaussian quadrature rules of Mousavi, Xiao & Sukumar [84] presented in Table 5.1 are compatible with the shape functions of Wachspress [36, 37] and Dasgupta [40] presented in §3.2.1.

node	ξ coordinate	η coordinate	weight
Exact for Polynomials of Degree 1			
1	0.0000000000000000	0.0000000000000000	2.3776412907378837
Exact for Polynomials of Degree 3			
1	-0.0349156305831802	0.6469731019095136	0.5449124407446143
2	-0.5951653065516678	-0.0321196846022659	0.6439082046243272
3	0.0349156305831798	-0.6469731019095134	0.5449124407446146
4	0.5951653065516677	0.0321196846022661	0.6439082046243275
Exact for Polynomials of Degree 5			
1	-0.0000000000000000	-0.0000000000000002	0.6257871064166934
2	-0.1351253857178451	0.7099621260052327	0.3016384608809768
3	-0.6970858746672087	0.1907259121533272	0.3169910433902452
4	-0.4651171392611024	-0.5531465782166917	0.3155445150066620
5	0.2842948078559476	-0.6644407817506509	0.2958801959111726
6	0.7117958231685716	-0.1251071394727008	0.2575426306970870
7	0.5337947578638855	0.4872045224587945	0.2642573384350463

Table 5.1: Generalized, Gaussian, quadrature, weights and nodes (a.k.a., cubature rules) for integrating over a regular pentagon in its natural coordinate system. These weights sum to the area of a pentagon inscribing an unit circle.

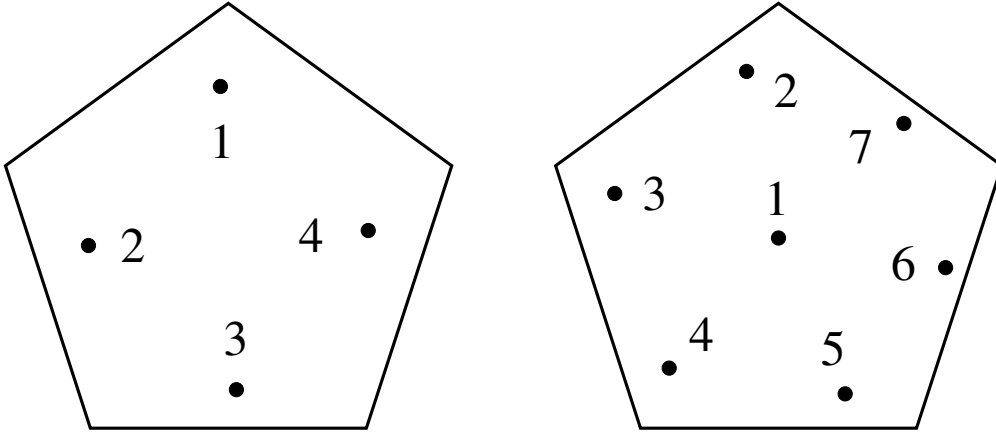


Figure 5.1: Locations of generalized, Gaussian, quadrature nodes for the 3rd (left) and 5th (right) degree integration methods presented in Table 5.1. Vertex 1 is located at the top of the pentagon, cf. Fig. 2.2, while the coordinate origin is located at its centroid (node 1 in the right figure).

Part 6

Variational Formulation

The problem we have set up to solve takes on the general form of

$$\mathbf{M}\ddot{\mathbf{x}} + \mathbf{K}\mathbf{x} = \mathbf{f}(t) \quad (6.1)$$

where \mathbf{M} is a mass matrix, \mathbf{K} is a stiffness matrix, \mathbf{f} is a forcing function, and \mathbf{x} is a displacement vector.

For our problem of interest,

$$\mathbf{x} = \sum_{v=1}^{20} \{x_v, y_v, z_v\}^T \quad (6.2a)$$

are the co-ordinates of vertex v located in the co-ordinate frame of the dodecahedron ($\mathbf{i}, \mathbf{j}, \mathbf{k}$) so that vectors \mathbf{f} and \mathbf{x} have length 60 while matrices \mathbf{M} and \mathbf{K} have dimension 60×60 with

$$\mathbf{M} = \mathbf{M}_{1D} + \mathbf{M}_{2D} + \mathbf{M}_{3D} \quad (6.2b)$$

$$\mathbf{K} = \mathbf{K}_{1D} + \mathbf{K}_{2D} + \mathbf{K}_{3D} \quad (6.2c)$$

$$\mathbf{f} = \mathbf{f}_{1D} + \mathbf{f}_{2D} + \mathbf{f}_{3D} \quad (6.2d)$$

where subscript ' $_{1D}$ ' applies to the alveolar chords, subscript ' $_{2D}$ ' applies to the alveolar septa, and subscript ' $_{3D}$ ' applies to the alveolar volume.

6.1. Mass Matrix

A consistent mass matrix [85] established in its natural co-ordinate system is defined as

$$\mathbf{M} = \sum_m \int_{V_m} \rho_m \mathbf{N}_m^T \mathbf{N}_m dV_m \quad (6.3)$$

wherein \mathbf{N}_m is the shape function matrix used to also construct the stiffness matrix for element m .

6.1.1. Mass Matrix of Chord

The determinant of the Jacobian matrix is used for the transformation of integral from the global coordinate system to the natural coordinate system by

$$|J| = \det J = \frac{\partial x}{\partial \xi} = \sum_{i=1}^n N_{i,\xi}(\xi) x_i \quad (6.4)$$

wherein N_i are the shape functions for a two-node alveolar chords in its natural coordinate system which are defined in a matrix form as

$$\mathbf{N} = \begin{bmatrix} \frac{1}{2} (1 - \xi) & \frac{1}{2} (1 + \xi) \end{bmatrix} \quad (6.5)$$

node	ξ coordinate	weight
	Exact for Polynomials of Degree 1	
1	0.00000000000000	2.00000000000000
	Exact for Polynomials of Degree 3	
1	-0.577350269189	1.00000000000000
2	0.577350269189	1.00000000000000
	Exact for Polynomials of Degree 5	
1	-0.774596669241	0.55555555555556
2	0.00000000000000	0.88888888888889
3	0.774596669241	0.55555555555556

Table 6.1: Generalized, Gaussian, quadrature, weights and nodes for integrating over a alveolar chord in its natural coordinate system.

wherein ξ is abscissae of the Gauss integration rule.

The consistent mass matrix of the 1-D alveolar chord that is evaluated numerically in its natural coordinate system can be described as

$$\mathbf{M}_{1C} = \int_{\Gamma} \rho \mathbf{N}^T \mathbf{N} A dx = \int_{-1}^1 \rho \mathbf{N}^T \mathbf{N} A |\mathbf{J}| d\xi = \sum_{i=1}^n \rho \mathbf{N}^T \mathbf{N} A |\mathbf{J}| w_i \quad (6.6)$$

with w_i being the weighting coefficients of the Gauss integration rule, and A being the cross section area of alveolar chord. Table 6.1 demonstrates the values of ξ and w_i for $n = 1, 2$, and 3 Gauss integration points.

The row-sum techniques is considered to construct the lumped mass matrix for 1-D alveolar chord, that is the sum of the elements of each row of the consistent mass matrix is used as the diagonal element [?]:

$$M_{1Lii} = \sum_{j=1}^n \int_{\Gamma} \rho N_i N_j A dx = \int_{-1}^1 \rho N_i A |\mathbf{J}| d\xi = \sum_{i=1}^n \rho N_i A |\mathbf{J}| w_i \quad (6.7)$$

wherein $\sum_{j=1}^n N_j = 1$.

The lumped-consistent weighted mass matrix \mathbf{M}_{1LC} of the 1-D alveolar chord is defined as

$$\mathbf{M}_{1LC} = (1 - \mu) \mathbf{M}_{1C} + \mu \mathbf{M}_{1L} \quad (6.8)$$

wherein μ is a free scalar parameter that is considered to be $\mu = 1/2$ to minimize low frequency dispersion.

For instance, the consistent mass matrix for alveolar chords with 1 Gauss integration point that is approximated by the weighted sum of function at the center of chord becomes

$$\mathbf{M}_{1C} = \frac{\rho A L}{4} \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} \quad (6.9)$$

where L is the length of alveolar chord. The row-sum techniques, gives the lumped mass matrix

$$\mathbf{M}_{1L} = \frac{\rho A L}{2} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad (6.10)$$

and the lumped-consistent weighted mass matrix is constructed as follow

$$\mathbf{M}_{1LC} = \frac{\rho A L}{8} \begin{bmatrix} 3 & 1 \\ 1 & 3 \end{bmatrix} \quad (6.11)$$

6.1.2. Mass Matrix of Pentagon

For the alveolar septa, the matrix of shape functions \mathbf{N} is arranged as

$$\mathbf{N} = \begin{bmatrix} N_1 & 0 & N_2 & 0 & N_3 & 0 & N_4 & 0 & N_5 & 0 \\ 0 & N_1 & 0 & N_2 & 0 & N_3 & 0 & N_4 & 0 & N_5 \end{bmatrix} \quad (6.12)$$

in which $N_i (i = 1, 2, 3, 4, 5)$ are five shape functions corresponding to the five vertices of the pentagon that are defined in Eq. (3.3). The consistent mass matrix \mathbf{M}_{2C} can also be obtained by substituting the above shape function matrix into

$$\mathbf{M}_{2C} = \int_V \rho \mathbf{N}^T \mathbf{N} dV = \int_{\diamond} \int_{\diamond} \rho \mathbf{N}^T \mathbf{N} |\mathbf{J}| h d\xi d\eta \quad (6.13)$$

wherein h being membrane thickness, and $|\mathbf{J}|$ being the determinant of the Jacobian matrix for pentagon. In quadrilateral derivations, the Jacobian of two-dimensional transformations that connect the x, y to ξ, η coordinate systems is needed. The components of Jacobian matrix are calculated using derivatives of shape functions with respect to the local and the current global coordinates at the i^{th} vertex via

$$[\mathbf{J}] = \begin{bmatrix} \partial x / \partial \xi & \partial y / \partial \xi \\ \partial x / \partial \eta & \partial y / \partial \eta \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^5 N_{i,\xi}(\xi, \eta) x_i & \sum_{i=1}^5 N_{i,\xi}(\xi, \eta) y_i \\ \sum_{i=1}^5 N_{i,\eta}(\xi, \eta) x_i & \sum_{i=1}^5 N_{i,\eta}(\xi, \eta) y_i \end{bmatrix} \quad (6.14)$$

The numerical integration of Eq. (6.13) result in

$$\mathbf{M}_{2C} = \sum_{p=1}^n \rho \mathbf{N}^T \mathbf{N} |\mathbf{J}| h w_i \quad (6.15)$$

where n stands for number of Gauss points, and w_i denotes the natural weight of the element. There are three mass matrix for each pentagon based upon three Gauss, quadrature rules in Table 5.1 which are integrating polynomials of order 3 and 5, respectively.

The row-sum techniques is considered to construct the Lumped mass matrix for pentagon, that makes the diagonal parameters as follow

$$M_{2Lii} = \sum_{j=1}^n \int_V \rho N_i N_j A dV = \int_{\diamond} \int_{\diamond} \rho N_i A |\mathbf{J}| d\xi d\eta = \sum_{i=1}^n \rho N_i h |\mathbf{J}| w_i \quad (6.16)$$

wherein $\sum_{j=1}^n N_j = 1$.

The lumped-consistent weighted mass matrix \mathbf{M}_{2LC} for the 2-D alveolar septa by choosing $\mu = 1/2$ is defined as

$$\mathbf{M}_{2LC} = (1 - \mu) \mathbf{M}_{2C} + \mu \mathbf{M}_{2L} = \frac{1}{2} (\mathbf{M}_{2C} + \mathbf{M}_{2L}) \quad (6.17)$$

For instance, the lumped-consistent mass matrix of a pentagon with 1 Gauss integration point at the center of pentagon that is constructed by averaging consistent mass matrix and the lumped mass matrix becomes

$$\mathbf{M}_{2LC} = \rho h \begin{bmatrix} 0.28532 & 0 & 0.04755 & 0 & 0.04755 & 0 & 0.04755 & 0 \\ 0.04755 & 0 & & & & & & \\ 0 & 0.28532 & 0 & 0.04755 & 0 & 0.04755 & 0 & 0.04755 \\ 0 & 0.04755 & & & & & & \\ 0.04755 & 0 & 0.28532 & 0 & 0.04755 & 0 & 0.04755 & 0 \\ 0.04755 & 0 & & & & & & \\ 0 & 0.04755 & 0 & 0.28532 & 0 & 0.04755 & 0 & 0.04755 \\ 0 & 0.04755 & & & & & & \\ 0.04755 & 0 & 0.04755 & 0 & 0.28532 & 0 & 0.04755 & 0 \\ 0.04755 & 0 & & & & & & \\ 0 & 0.04755 & 0 & 0.04755 & 0 & 0.28532 & 0 & 0.04755 \\ 0 & 0.04755 & & & & & & \\ 0.04755 & 0 & 0.04755 & 0 & 0.04755 & 0 & 0.28532 & 0 \\ 0.04755 & 0 & & & & & & \\ 0 & 0.04755 & 0 & 0.04755 & 0 & 0.04755 & 0 & 0.28532 \\ 0 & 0.04755 & & & & & & \\ 0.04755 & 0 & 0.04755 & 0 & 0.04755 & 0 & 0.04755 & 0 \\ 0.28532 & 0 & 0.04755 & & & & & \\ 0 & 0.04755 & 0 & 0.04755 & 0 & 0.04755 & 0 & 0.04755 \\ 0 & 0.28532 & & & & & & \end{bmatrix} \quad (6.18)$$

6.1.3. Mass Matrix of Tetrahedron

The dodecahedron has 60 individual tetrahedral whereas its origin being the common vertex of all tetrahedrons. Hence, the analysis to find the mass matrix of a tetrahedron is used to reach the mass matrix of whole alveolar volume.

The matrix of shape functions \mathbf{N} for a tetrahedon has the form of

$$\mathbf{N} = \begin{bmatrix} N_1 & 0 & 0 & N_2 & 0 & 0 & N_3 & 0 & 0 & N_4 & 0 & 0 \\ 0 & N_1 & 0 & 0 & N_2 & 0 & 0 & N_3 & 0 & 0 & N_4 & 0 \\ 0 & 0 & N_1 & 0 & 0 & N_2 & 0 & 0 & N_3 & 0 & 0 & N_4 \end{bmatrix} \quad (6.19)$$

in which $N_i (i = 1, 2, 3, 4)$ are four shape functions corresponding to the four vertices of the tetrahedron that are defined as follow

$$N_1 = 1 - \xi - \eta - \zeta \quad (6.20a)$$

$$N_2 = \xi \quad (6.20b)$$

$$N_3 = \eta \quad (6.20c)$$

$$N_4 = \zeta \quad (6.20d)$$

The numerical integration is used to obtain the mass matrix of tetrahedron via

$$\mathbf{M}_{3D} = \sum_{p=1}^{n_p} \rho \mathbf{N}^T \mathbf{N} |\mathbf{J}| w_p \quad (6.21)$$

wherein $|\mathbf{J}|$ being the determinant of the Jacobian matrix in a tetrahedron that are calculated using derivatives of shape functions with respect to the local coordinates (ξ, η, ζ) , and the current global coordinates (x_i, y_i, z_i) at the i^{th} vertex via

$$\begin{aligned} [\mathbf{J}] &= \begin{bmatrix} \partial x / \partial \xi & \partial y / \partial \xi & \partial z / \partial \xi \\ \partial x / \partial \eta & \partial y / \partial \eta & \partial z / \partial \eta \\ \partial x / \partial \zeta & \partial y / \partial \zeta & \partial z / \partial \zeta \end{bmatrix} \\ &= \begin{bmatrix} \sum_{i=1}^4 N_{i,\xi}(\xi, \eta, \zeta) x_i & \sum_{i=1}^4 N_{i,\xi}(\xi, \eta, \zeta) y_i & \sum_{i=1}^4 N_{i,\xi}(\xi, \eta, \zeta) z_i \\ \sum_{i=1}^4 N_{i,\eta}(\xi, \eta, \zeta) x_i & \sum_{i=1}^4 N_{i,\eta}(\xi, \eta, \zeta) y_i & \sum_{i=1}^4 N_{i,\eta}(\xi, \eta, \zeta) z_i \\ \sum_{i=1}^4 N_{i,\zeta}(\xi, \eta, \zeta) x_i & \sum_{i=1}^4 N_{i,\zeta}(\xi, \eta, \zeta) y_i & \sum_{i=1}^4 N_{i,\zeta}(\xi, \eta, \zeta) z_i \end{bmatrix} \end{aligned} \quad (6.22)$$

There are three mass matrix for each tetrahedron based upon three Gauss, quadrature rules in Table 6.2 which are integrating polynomials of order 1, 2 and 3, respectively.

The row-sum techniques is considered to construct the lumped mass matrix for pentagon, that makes the diagonal parameters as follow

$$M_{3Lii} = \sum_{j=1}^n \int_V \rho N_i N_j A dV = \sum_{i=1}^n \rho N_i |\mathbf{J}| w_i \quad (6.23)$$

wherein $\sum_{j=1}^n N_j = 1$.

The lumped-consistent weighted mass matrix \mathbf{M}_{3LC} for the 3-D alveolar volume by choosing $\mu = 1/2$ becomes

$$\mathbf{M}_{3LC} = (1 - \mu) \mathbf{M}_{3C} + \mu \mathbf{M}_{3L} = \frac{1}{2} (\mathbf{M}_{3C} + \mathbf{M}_{3L}) \quad (6.24)$$

node	ξ coordinate	η coordinate	ζ coordinate	weight
Exact for Polynomials of Degree 1				
1	1/4	1/4	1/4	1/6
Exact for Polynomials of Degree 2				
1	$(5 - \sqrt{5})/20$	$(5 - \sqrt{5})/20$	$(5 - \sqrt{5})/20$	1/24
2	$(5 - \sqrt{5})/20$	$(5 - \sqrt{5})/20$	$(5 + 3\sqrt{5})/20$	1/24
3	$(5 - \sqrt{5})/20$	$(5 + 3\sqrt{5})/20$	$(5 - \sqrt{5})/20$	1/24
4	$(5 + 3\sqrt{5})/20$	$(5 - \sqrt{5})/20$	$(5 - \sqrt{5})/20$	1/24
Exact for Polynomials of Degree 3				
1	1/4	1/4	1/4	-2/15
2	1/6	1/6	1/6	3/40
3	1/6	1/6	1/2	3/40
4	1/6	1/2	1/6	3/40
5	1/2	1/6	1/6	3/40

Table 6.2: Generalized, Gaussian, quadrature, weights and nodes for integrating over a tetrahedron in its natural coordinate system.

References

- [1] J. D. Clayton, A. D. Freed, A constitutive framework for finite viscoelasticity and damage based on the Gram-Schmidt decomposition, in review (2019).
- [2] J. D. Stitzel, F. S. Gayzik, J. J. Hoth, J. Mercier, H. D. Gage, K. A. Morton, S. M. Duma, R. M. Payne, Development of a finite element-based injury metric for pulmonary contusion, Part I: Model development and validation, *Stapp Car Crash Journal* 49 (2005) 271–289.
- [3] J. H. Stuhmiller, C. J. Chuong, Y. Y. Phillips, K. T. Dodd, Computer modeling of thoracic response to blast, *Journal of Trauma* 28 (1988) S132–S139.
- [4] A. A. Vlessis, D. D. Trunkey, Non-penetrating injury of the thorax, in: G. J. Cooper, H. A. F. Dudley, D. S. Gann, R. A. Little, R. L. Maynard (Eds.), *Scientific Foundations of Trauma*, Butterworth Heinemann, Oxford, UK, 1997, pp. 127–143.
- [5] Y.-C. Fung, P. Patitucci, P. Tong, Stress and strain in the lung, *ASCE Journal of Engineering Mechanics* 104 (1978) 201–223.
- [6] D. L. Vawter, Y.-C. Fung, J. B. West, Constitutive equation of lung tissue elasticity, *Journal of Biomechanical Engineering* 101 (1979) 38–45.
- [7] D. L. Vawter, A finite element model for macroscopic deformation of the lung, *Journal of Biomechanical Engineering* 102 (1980) 1–7.
- [8] F. S. Gayzik, J. J. Hoth, M. Daly, J. W. Meredith, J. D. Stitzel, A finite element based injury metric for pulmonary contusion: investigation of candidate metrics through correlation with computed tomography, *Stapp Car Crash Journal* 51 (2007) 189–209.
- [9] F. S. Gayzik, J. J. Hoth, J. Stitzel, Finite element-based injury metrics for pulmonary contusion via concurrent model optimization, *Biomechanics and Modeling in Mechanobiology* 10 (2011) 505–520.
- [10] K. Yuen, D. S. Cronin, Y. C. Deng, Lung response and injury in side impact conditions, in: *Proceedings of the 2008 International Research Council on Biomechanics of Injury (IRCOBI) Conference*, 2008.
- [11] A. D. Freed, D. R. Einstein, J. P. Carson, R. E. Jacob, Viscoelastic model for lung parenchyma for multi-scale modeling of respiratory system, Phase II: Dodecahedral micro-model, Tech. Rep. PNNL-21287, Pacific Northwest National Laboratory, Richland, WA (March 2012).
- [12] J. P. Butler, H. Miki, S. Squarcia, R. A. Rogers, J. L. Lehr, Effect of macroscopic deformation on lung microstructure, *Journal of Applied Physiology* 81 (1996) 1792–1799.
- [13] E. Roan, C. M. Waters, What do we know about mechanical strain in lung alveoli?, *American Journal of Physiology—Lung Cellular and Molecular Physiology* 301 (2011) L625–L635.
- [14] Y. C. Fung, *Biomechanics: Motion, flow, stress, and growth*, Springer-Verlag, New York, 1990.
- [15] Y. C. Fung, Stress, deformation, and atelectasis of the lung, *Circulation Research* 37 (1975) 481–496.
- [16] Y. C. Fung, A model of the lung structure and its validation, *Journal of Applied Physiology* 64 (1988) 2132–2141.
- [17] F. G. Hoppin, Jr., J. Hildebrandt, Mechanical properties of the lung, in: J. B. West (Ed.), *Bioengineering Aspects of the Lung*, Vol. 3 of *Lung Biology in Health and Disease*, Marcel Dekker, New York, 1977, pp. 83–162.
- [18] A. Frankus, G. C. Lee, A theory for distortion studies of lung parenchyma based on alveolar membrane properties, *Journal of Biomechanics* 7 (1974) 101–107.
- [19] J. de Ryk, J. Thiesse, E. Namati, G. McLennan, Stress distribution in a three dimensional, geometric alveolar sac under normal and emphysematous conditions, *International Journal of COPD* 2 (2007) 81–91.
- [20] P. J. Dale, F. L. Matthews, R. C. Schroter, Finite element analysis of lung alveolus, *Journal of Biomechanics* 13 (1980) 865–873.
- [21] E. Denny, R. C. Schroter, The mechanical behavior of a mammalian lung alveolar duct model, *Journal of Biomechanical Engineering* 117 (1995) 254–261.
- [22] E. Denny, R. C. Schroter, Relationships between alveolar size and fibre distribution in mammalian lung alveolar duct model, *Journal of Biomechanical Engineering* 119 (1997) 289–297.
- [23] E. Denny, R. C. Schroter, Viscoelastic behavior of a lung alveolar duct model, *Journal of Biomechanical Engineering* 122 (2000) 143–151.

- [24] R. Kowe, R. C. Schroter, F. L. Matthews, D. Hitchings, Analysis of elastic and surface tension effects in the lung alveolus using finite element methods, *Journal of Biomechanics* 19 (7) (1986) 541–549.
- [25] Z. l. Chen, Y. z. Chen, Z. y. Hu, A micromechanical model for estimating alveolar wall strain in mechanically ventilated edematous lungs, *Journal of Applied Physiology* 117 (2014) 586–592.
- [26] E. Kimmel, R. D. Kamm, A. H. Shapiro, A cellular model of lung elasticity, *Journal of Biomechanical Engineering* 109 (1987) 126–131.
- [27] B. Budiansky, E. Kimmel, Elastic moduli of lungs, *Journal of Applied Mechanics* 54 (1987) 351–358.
- [28] E. Kimmel, B. Budiansky, Surface tension and the dodecahedron model for lung elasticity, *Journal of Biomechanical Engineering* 112 (1990) 160–167.
- [29] B. Weed, S. Patnaik, M. Rougeau-Browning, B. Brazile, J. Liao, R. Prabhu, L. N. Williams, Experimental evidence of mechanical isotropy in porcine lung parenchyma, *Materials* 8 (2015) 2454–2466.
- [30] J. M. B. Hughes, F. G. Hoppin, Jr., J. Mead, Effect of lung inflation on bronchial length and diameter in excised lungs, *Journal of Applied Physiology* 32 (1) (1972) 25–35.
- [31] J. Mead, Respiration: pulmonary mechanics, *Annual Review of Physiology* 35 (1973) 169–192.
- [32] J. B. West, *Respiratory Physiology: The essentials*, eighth Edition, Lippincott Williams & Wilkins, Philadelphia, 2007.
- [33] E. Denny, R. C. Schroter, A model of non-uniform lung parenchyma distortion, *Journal of Biomechanics* 39 (2006) 652–663.
- [34] S. S. Sobin, Y. C. Fung, H. M. Tremer, Collagen and elastin fibers in human pulmonary alveolar walls, *Journal of Applied Physiology* 64 (4) (1988) 1659–1675.
- [35] A. D. Freed, J. B. le Graverend, K. R. Rajagopal, A technical note: An elastic-inelastic decomposition of Laplace stretch, *Acta Mechanica* In press. doi:10.1007/s00707-019-02462-3.
- [36] E. L. Wachspress, *A Rational Finite Element Basis*, Academic Press, New York, 1975.
- [37] E. Wachspress, *Rational Bases and Generalized Barycentrics: Applications to finite elements and graphics*, Springer, Cham, 2016.
- [38] T. J. R. Hughes, *The Finite Element Method: Linear static and dynamic finite element analysis*, Prentice-Hall, Englewood Cliffs, N.J., 1987.
- [39] N. Sukumar, E. A. Malsch, Recent advances in the construction of polygonal finite element interpolants, *Archives of Computational Methods in Engineering* 13 (2006) 129–163.
- [40] G. Dasgupta, Interpolants within convex polygons: Wachspress shape functions, *Journal of Aerospace Engineering* 16 (2003) 1–8.
- [41] J. D. Clayton, *Differential Geometry and Kinematics of Continua*, World Scientific, New York, 2015.
- [42] A. R., Srinivasa, On the use of the upper triangle (or **QR**) decomposition for developing constitutive equations for Green-elastic materials, *International Journal of Engineering Science* 60 (2012) 1–12.
- [43] A. D. Freed, A. R., Srinivasa, Logarithmic strain and its material derivative for a **QR** decomposition of the deformation gradient, *ACTA Mechanica* 226 (2015) 2645–2670. doi:10.1007/s00707-015-1344-0.
- [44] A. D. Freed, V. Erel, M. R. Moreno, Conjugate stress/strain base pairs for planar analysis of biological tissues, *Journal of Mechanics of Materials and Structures* 12 (2017) 219–247.
- [45] A. D. Freed, S. Zamani, Elastic Kelvin-Poisson-Poynting solids described through scalar conjugate stress/strain pairs derived from a **QR** factorization of **F**, *Journal of the Mechanics and Physics of Solids* doi:10.1016/j.jmps.2019.05.011.
- [46] A. G. McLellan, Finite strain coordinate and the stability of solid phases, *Journal of Physics C: Solid State Physics* 9 (1976) 4083–4094.
- [47] A. G. McLellan, *The Classical Thermodynamics of Deformable Materials*, Cambridge Monographs in Physics, Cambridge University Press, Cambridge, 1980.
- [48] P. Rosakis, Ellipticity and deformations with discontinuous gradients in finite elastostatics, *Archives for Rational Mechanics and Analysis* 109 (1990) 1–37.
- [49] R. Souchet, Concerning the polar decomposition of the deformation gradient, *International Journal of Engineering Science* 31 (1993) 1499–1506.
- [50] A. D. Freed, S. Zamani, On the use of convected coordinate systems in the mechanics of continuous media derived from a **QR** factorization of **F**, *International Journal of Engineering Science* 127 (2018)

- 145–161. doi:10.1016/j.ijengsci.2018.02.011.
- [51] A. D. Freed, A note on stress/strain conjugate pairs: explicit and implicit theories of thermoelasticity for anisotropic materials, *International Journal of Engineering Science* 120 (2017) 155–171. doi:10.1016/j.ijengsci.2017.08.002.
 - [52] C. E. Perlman, J. Bhattacharya, Alveolar expansion imaged by optical sectioning microscopy, *Journal of Applied Physiology* 103 (2007) 1037–1044.
 - [53] L. R. G. Treloar, *The Physics of Rubber Elasticity*, 3rd Edition, Clarendon Press, Oxford, 1975.
 - [54] B. Suki, S. Ito, D. Stamenović, K. R. Lutchen, E. P. Ingenito, Biomechanics of the lung parenchyma: critical roles of collagen and mechanical forces, *Journal of Applied Physiology* 98 (2005) 1892–1899.
 - [55] B. Suki, D. Stamenović, R. Hubmayr, Lung parenchymal mechanics, *Comprehensive Physiology* 1 (2011) 1317–1351.
 - [56] C. Carathéodory, Untersuchungen über die Grundlagen der Thermodynamik, *Mathematische Annalen* 67 (1909) 355–386, translated in: J. Kestin (ed.), *The Second Law of Thermodynamics*, Dowden, Hutchinson and Ross, Stroudsburg, PA, 1976, pp. 229–256.
 - [57] F. Weinhold, Metric geometry of equilibrium thermodynamics. III. Elementary formal structure of a vector-algebraic representation of equilibrium thermodynamics, *The Journal of Chemical Physics* 63 (1975) 2488–2495.
 - [58] R. Gilmore, Length and curvature in the geometry of thermodynamics, *Physical Review A* 30 (1984) 1994–1997.
 - [59] C. Truesdell, Hypoelasticity, *Journal of Rational Mechanics and Analysis* 4 (1955) 83–133.
 - [60] H. Saraf, K. T. Ramesh, A. M. Lennon, A. C. Merkle, J. C. Roberts, Mechanical properties of soft human tissues under dynamic loading, *Journal of Biomechanics* 40 (2007) 1960–1967.
 - [61] G. F. Becker, The finite elastic stress-strain function, *American Journal of Science* 46 (1893) 337–356.
 - [62] M. A. Hajji, T. A. Wilson, S. J. Lai-Fook, Improved measurements of shear modulus and pleural membrane tension of the lung, *Journal of Applied Physiology* 47 (1979) 175–181.
 - [63] Ames Research Staff, Equations, tables, and charts for compressible flow, Tech. Rep. NACA 1135, National Advisory Committee for Aeronautics (1953).
 - [64] D. Stamenović, Micromechanical foundations of pulmonary elasticity, *Physiological Reviews* 70 (4) (1990) 1117–1134.
 - [65] B. A. Hills, An alternative view of the role(s) of surfactant and the alveolar model, *Journal of Applied Physiology* 87 (1999) 1567–1583.
 - [66] J. H. T. Bates, A recruitment model of quasi-linear power-law stress adaptation in lung tissue, *Annals of Biomedical Engineering* 35 (2007) 1165–1174.
 - [67] M. Matsuda, Y. C. Fung, S. S. Sobin, Collagen and elastin fibers in human pulmonary alveolar mouths and ducts, *Journal of Applied Physiology* 63 (3) (1987) 1185–1194.
 - [68] A. D. Freed, K. R. Rajagopal, A promising approach for modeling biological fibers, *ACTA Mechanica* 227 (2016) 1609–1619, errata: **229** (2018), pg. 3573. doi:10.1007/s00707-016-1583-8; 10.1007/s00707-018-2183-6.
 - [69] J. Kastelic, A. Galeski, E. Baer, The multicomposite structure of tendon, *Connective Tissue Research* 6 (1978) 11–23.
 - [70] A. D. Freed, T. C. Doehring, Elastic model for crimped collagen fibrils, *Journal of Biomechanical Engineering* 127 (2005) 587–593.
 - [71] B. B. Aaron, J. M. Gosline, Elastin as a random-network elastomer: a mechanical and optical analysis of single elastin fibers, *Biopolymers* 20 (1981) 1247–1260.
 - [72] D. W. Urry, Physicochemical properties of elastin and constituent peptides, Vol. 1 of *Elastin and Elastases*, CRC Press, Boca Raton, 1989, pp. 141–173.
 - [73] I. G. Fels, Hydration and density of collagen and gelatin, *Journal of Applied Polymer Science* 8 (1964) 1813–1824.
 - [74] J. R. Kanagy, Specific heats of collagen and leather, *Journal of Research of the National Bureau of Standards* 55 (1955) 191–195.
 - [75] C. E. Weir, Effect of temperature on the volume of leather and collagen in water, *Journal of Research*

- of the National Bureau of Standards 41 (1948) 279–285.
- [76] M. A. Lillie, J. M. Gosline, Unusual swelling of elastin, *Biopolymers* 64 (2002) 115–126.
 - [77] R. E. Shadwick, J. M. Gosline, Physical and chemical properties of rubber-like elastic fibres from the octopus aorta, *Journal of Experimental Biology* 114 (1985) 239–257.
 - [78] S. R. Kakivaya, C. A. J. Hoeve, The glass point of elastin, *Proceedings of the National Academy of Sciences, USA* 72 (1975) 3505–3507.
 - [79] M. A. Lillie, J. M. Gosline, Mechanical properties of elastin along the thoracic aorta in the pig, *Journal of Biomechanics* 40 (2007) 2214–2221.
 - [80] H. Hörmann, H. Schleich, Reversible and irreversible denaturation of collagen fibers, *Biochemistry* 10 (6) (1971) 932–937.
 - [81] L. Davison, *Fundamentals of Shock Wave Propagation in Solids, Shock Wave and High Pressure Phenomena*, Springer, Berlin, 2008.
 - [82] A. D. Freed, A technical note: Two-step PECE methods for approximating solutions to first- and second-order ODEs, Tech. Rep. arXiv:1707.02125 [cs.NA], Texas A&M University (2017).
 - [83] G. Söderlind, Automatic control and adaptive time-stepping, *Numerical Algorithms* 31 (2002) 281–310.
 - [84] S. E. Mousavi, H. Xiao, N. Sukumar, Generalized Gaussian quadrature rules on arbitrary polygons, *International Journal for Numerical Methods in Engineering* 82 (2010) 99–113.
 - [85] J. S. Archer, Consistent matrix formulations for structural analysis using finite-element techniques, *AIAA Journal* 3 (1965) 1910–1918.
 - [86] G. Green, On the propagation of light in crystallized media, *Transactions of the Cambridge Philosophical Society* 7 (1841) 121–140.
 - [87] K. R. Rajagopal, On implicit constitutive theories, *Applications of Mathematics* 48 (4) (2003) 279–319.
 - [88] A. R. Akintunde, K. S. Miller, Evaluation of microstructurally motivated constitutive models to describe age-dependent tendon healing, *Biomechanics and Modeling in Mechanobiology* 17 (2018) 793–814. doi:10.1007/s10237-017-0993-4.

Appendix A. Implicit Elasticity

Both explicit (Green [86] elastic) and implicit (Rajagopal [87] elastic) material models are put forward in this appendix for one's consideration when choosing a material model to represent biologic fibers and membranes. We discuss thermoelastic fibers first, and then thermoelastic membranes. We have no need to address thermoelastic volumes for our application beyond what has been presented in §4.3.4. We employ Gibbs' free-energy potentials instead of internal-energy potentials to the extent that is possible. They relate to one another via a well-known Legendre transformation. A Gibbs energy approach implies that a change in the intensive variables (thermodynamic forces) will cause a response in the extensive variables (thermodynamic displacements), which is the exact opposite of an internal energy approach. Cause and effect are correct in a Gibbs formulation.

Appendix A.1. Alveolar Chords as Green (Explicit) Thermoelastic Fibers

For a 1D fiber with a mass density of ρ per unit length, the conjugate fields are: temperature θ and entropy η , plus force F and strain $e := \ln(L/L_0)$ with L_0 and L denoting the initial and current fiber lengths. A Green elastic fiber adopts a Gibbs free-energy function with an explicit dependence upon state, viz., $\mathcal{G}(\theta, F)$ such that $d\mathcal{G} = -\eta d\theta - \frac{1}{\rho}e dF$ (cf. Eqn. 4.1a), from which one derives its governing constitutive equations as being

$$\eta = -\partial_\theta \mathcal{G}(\theta, F) \quad \text{and} \quad e = -\rho \partial_F \mathcal{G}(\theta, F) \quad (\text{A.1a})$$

that, when differentiated, can be rearranged into the following hypo-elastic equation

$$\begin{Bmatrix} d\eta \\ de \end{Bmatrix} = - \begin{bmatrix} \partial_{\theta\theta} \mathcal{G} & \partial_{\theta F} \mathcal{G} \\ \rho \partial_{F\theta} \mathcal{G} & \rho \partial_{FF} \mathcal{G} \end{bmatrix} \begin{Bmatrix} d\theta \\ dF \end{Bmatrix} = \begin{bmatrix} C/\theta & \alpha/\rho \\ \alpha & 1/E \end{bmatrix} \begin{Bmatrix} d\theta \\ dF \end{Bmatrix} \quad (\text{A.1b})$$

whose thermo-physical material properties are, in general, functions of state defined by

$$C := \theta \partial_\theta \eta|_F = -\theta \partial_{\theta\theta} \mathcal{G}(\theta, F) \quad (\text{A.1c})$$

$$\alpha := L^{-1} \partial_\theta L|_F = \partial_\theta e|_F = -\rho \partial_{F\theta} \mathcal{G}(\theta, F) \equiv -\rho \partial_{\theta F} \mathcal{G}(\theta, F) \quad (\text{A.1d})$$

$$1/E := L^{-1} \partial_F L|_\theta = \partial_F e|_\theta = -\rho \partial_{FF} \mathcal{G}(\theta, F) \quad (\text{A.1e})$$

where the elastic compliance $1/E = L^{-1} \partial_F L|_\theta = \partial_F \ln(L/L_0)|_\theta = \partial_F e|_\theta$ has units of reciprocal force evaluated at constant temperature, while the thermal expansion coefficient $\alpha = L^{-1} \partial_\theta L|_F = \partial_\theta \ln(L/L_0)|_F = \partial_\theta e|_F$ has units of reciprocal temperature evaluated at constant force. The mass density ρ is one-dimensional in this presentation, i.e., it has units of mass per unit length of fiber; likewise, modulus E has units of force, not stress.

Appendix A.1.1. A Hookean Fiber

A thermoelastic Hookean fiber is Green elastic with a Gibbs free energy described by

$$\mathcal{G}(\theta, F) = -C \left(\theta \ln \left(\frac{\theta}{\theta_0} \right) - (\theta - \theta_0) \right) - \frac{F}{\rho} \left(\alpha(\theta - \theta_0) + \frac{F}{2E} \right) \quad (\text{A.2})$$

which is a function of temperature θ and force F , normalized so that $\mathcal{G}(\theta_0, 0) = 0$, and is compatible with the formulation expressed in Eqn. (A.1). In this model the material constants ρ , C , α and E are all considered to be of constant value across state space.

Appendix A.2. Alveolar Chords as Rajagopal (Implicit) Thermoelastic Fibers

In 2003, Rajagopal [87] introduced the idea of an implicit elastic solid. In 2016, Freed & Rajagopal [68] constructed an elastic fiber model that convolves an explicit energy with an implicit energy. In their approach they decomposed strain $e := \ln(L/L_0)$ into a sum of two strains, viz., $e = e_1 + e_2$ where $e_1 := \ln(L_1/L_0)$ and $e_2 := \ln(L/L_1)$. Length L_0 is an initial fiber length, viz., its length whenever $F = 0$. Length L_1 can be thought of as a fiber length caused solely by an unraveling of molecular configuration (e.g., an unraveling of collagen crimp) under an applied load of F . The state associated with length L_1 is non-physical in that one cannot unravel molecules without also stretching them to some extent. Final length L is the actual fiber length under an applied load F caused by both a reconfiguration and a stretching of its molecular network. Here we present their ideas in terms of a Gibbs free-energy function. (Freed & Rajagopal originally used a Helmholtz free-energy function.)

Let the Gibbs free energy be described by a function of the form⁸

$$\mathcal{G}(\theta, e, F) := \mathcal{G}_1(e_1, F) + \mathcal{G}_2(\theta, F) \quad \text{where} \quad d\mathcal{G} = -\eta d\theta - \frac{1}{\rho} e dF \quad (\text{A.3})$$

with \mathcal{G}_1 being an implicit potential (a configuration energy) and \mathcal{G}_2 being an explicit potential (a strain energy). This energy function leads to the same constitutive equation displayed in Eqn. (A.1b), but whose material properties (A.1c–A.1e) are now interpreted according to the expressions

$$C := \theta \partial_\theta \eta|_F = -\theta \partial_{\theta\theta} \mathcal{G}_2(\theta, F) \quad (\text{A.4a})$$

$$\alpha := \partial_\theta e|_F = -\rho \partial_{F\theta} \mathcal{G}_2(\theta, F) \equiv -\rho \partial_{\theta F} \mathcal{G}_2(\theta, F) \quad (\text{A.4b})$$

$$1/E := \partial_F e|_\theta = -(\rho \partial_{e_1} \mathcal{G}_1(e_1, F))^{-1} (e + \rho \partial_F \mathcal{G}(\theta, e, F)) - \rho \partial_{FF} \mathcal{G}_2(\theta, F) \quad (\text{A.4c})$$

whose elastic compliance $1/E$ is found to be the sum of two compliances: one explicit in origin and the other implicit in origin, and where the mass density ρ is per unit length.

Derivation: Because Gibbs energy is a state function, its differential is exact allowing one to write the left-hand side of the thermodynamic expression $d\mathcal{G} = -\eta d\theta - \frac{1}{\rho} e dF$ as $d\mathcal{G} = \partial_{e_1} \mathcal{G}_1 de_1 + \partial_F \mathcal{G}_1 dF + \partial_\theta \mathcal{G}_2 d\theta + \partial_F \mathcal{G}_2 dF$. Recalling that $e = e_1 + e_2$, the explicit (hyperelastic) terms combine to produce constitutive expressions of

$$\eta = -\partial_\theta \mathcal{G}_2(\theta, F) \quad \text{and} \quad e_2 = -\rho \partial_F \mathcal{G}_2(\theta, F)$$

with the remaining implicit terms collecting to produce a differential equation of

$$\rho \partial_{e_1} \mathcal{G}_1(e_1, F) de_1 = -(e_1 + \rho \partial_F \mathcal{G}_1(e_1, F)) dF.$$

Differentiating the constitutive equation for entropy with respect to state leads directly to the expressions for specific heat C and thermal expansion α stated in Eqns. (A.4a &

⁸One might be tempted to consider an implicit energy function of the form $\mathcal{G} = \mathcal{G}_1(\theta, e_1, F) + \mathcal{G}_2(\theta, F)$, but this would lead to a non-symmetric susceptibility matrix. Consequently, it would not satisfy Maxwell's thermodynamic constraint for integrability; hence, it is inadmissible as a Gibbs potential.

A.4b). Recalling that the strains add, i.e., $e = e_1 + e_2$, and therefore so do their rates, viz., $de = de_1 + de_2$, a consequence of them being logarithmic in construction, then rearranging the implicit constitutive equation to solve for de_1 , while differentiating the explicit constitutive equation for e_2 , and finally adding these strain increments to get de , results in the elastic compliance function stated in Eqn. (A.4c). \square

Appendix A.2.1. A Biologic Fiber

The fiber model of Freed & Rajagopal [68] imposes a strain-limiting constraint onto internal strain e_1 whenever one considers a Gibbs free-energy function of the form

$$\mathcal{G}_1(e_1, F) = -\frac{1}{\rho} \left(e_t(E_1 e_1 - F) + 2e_1 F \right) \quad (\text{A.5a})$$

$$\mathcal{G}_2(\theta, F) = -C \left(\theta \ln \left(\frac{\theta}{\theta_0} \right) - (\theta - \theta_0) \right) - \frac{F}{\rho} \left(\alpha(\theta - \theta_0) + \frac{F}{2E_2} \right) \quad (\text{A.5b})$$

which depend upon temperature θ , force F , and internal strain e_1 , normalized so that $\mathcal{G}_1(0, 0) = 0$ and $\mathcal{G}_2(\theta_0, 0) = 0$. In fact, the explicit energy adopted here is Hookean, cf. Eqn. (A.2). The resulting constitutive responses for entropy η and strain $e = \ln(L/L_0)$ are therefore described by the following matrix differential equation

$$\begin{Bmatrix} d\eta \\ de \end{Bmatrix} = \begin{bmatrix} C/\theta & \alpha/\rho \\ \alpha & 1/E \end{bmatrix} \begin{Bmatrix} d\theta \\ dF \end{Bmatrix} \quad (\text{A.5c})$$

with an elastic compliance whose tangent response is described by

$$\frac{1}{E(\theta, e, F)} = \frac{e_t - e_1}{E_1 e_t + 2F} + \frac{1}{E_2} \quad \text{wherein} \quad e_1 = e - \alpha(\theta - \theta_0) - \frac{F}{E_2} \quad (\text{A.5d})$$

with an initial tangent modulus $E(\theta_0, 0, 0)$ of $E_1 E_2 / (E_1 + E_2) \approx E_1$ whenever $E_2 \gg E_1 > 0$, while the terminal tangent modulus $E(e_1 = e_t)$ is E_2 , with a transition strain occurring at $e_t > 0$ that establishes a limiting state for internal strain e_1 , i.e., $0 \leq e_1 < e_t$, which is that strain whereat a fiber's molecular configuration has been completely unraveled. Here ρ is a mass per unit length of fiber, C is a specific heat per unit mass at constant force, and α is a coefficient of linear thermal expansion at constant force, all of which have the same physical interpretation as their counterparts for the Hookean fiber. Only their elastic compliances are interpreted differently. This model has been found to be superior to other models commonly employed in the literature for modeling biologic fibers [88].

Biologic fibers, per our application, are long and slender. Consequently, they will buckle under compression. Buckling is not accounted for in our modeling of alveolar chords. Rather, it is assumed that the compliant response at the origin, with modulus $E_1 E_2 / (E_1 + E_2)$, continues into compression, thereby ensuring a measure of numeric stability in our software.

The above methodology would allow us to construct a suite of thermodynamically admissible elastic compliance functions, but we only have need for the one considered above.

Appendix A.3. Alveolar Septa as Green (Explicit) Thermoelastic Membranes

We observed in §4.2 that an alveolar membrane has a response comprised of an uniform contribution and a non-uniform contribution, and that these two contributions are uncoupled; consequently, their internal energies add in that $\mathcal{U}(\eta, \xi, \varepsilon, \gamma) = \mathcal{U}_u(\eta, \xi) + \mathcal{U}_n(\varepsilon, \gamma)$ with \mathcal{U}_u being the uniform contribution of \mathcal{U} and \mathcal{U}_n being the non-uniform contribution of \mathcal{U} . It is advantageous to relate the material constants to a Gibbs free-energy approach for the uniform contribution, while retaining the internal energy approach for its non-uniform contribution.

Appendix A.3.1. Uniform Response

From thermodynamics, Eqn. (4.1b), comes $d\mathcal{U}_u = \theta d\eta + \frac{1}{\rho}T dA/A = \theta d\eta + \frac{1}{\rho}\pi d\xi$ where $\pi = 2T$ and $d\xi = \frac{1}{2}A^{-1}dA$ whose mass density ρ now has units of mass per unit area. Upon writing this expression in its Gibbs form $d\mathcal{G}_u = -\eta d\theta - \frac{1}{\rho}\xi d\pi$ via a Legendre transformation comes a constitutive equation appropriate for describing the uniform response of a thermoelastic planar membrane, namely

$$\begin{Bmatrix} d\eta \\ d\xi \end{Bmatrix} = - \begin{bmatrix} \partial_{\theta\theta}\mathcal{G}_u & \partial_{\theta\pi}\mathcal{G}_u \\ \rho \partial_{\pi\theta}\mathcal{G}_u & \rho \partial_{\pi\pi}\mathcal{G}_u \end{bmatrix} \begin{Bmatrix} d\theta \\ d\pi \end{Bmatrix} = \begin{bmatrix} C/\theta & \alpha/2\rho \\ \alpha/2 & 1/4M \end{bmatrix} \begin{Bmatrix} d\theta \\ d\pi \end{Bmatrix} \quad (\text{A.6a})$$

with material constants defined accordingly

$$C := \theta \partial_{\theta\eta}|_{\pi} = -\theta \partial_{\theta\theta}\mathcal{G}_u \quad (\text{A.6b})$$

$$\alpha := A^{-1} \partial_{\theta}A|_T = 2 \partial_{\theta}\xi|_{\pi} = -2\rho \partial_{\pi\theta}\mathcal{G}_u = -2\rho \partial_{\theta\pi}\mathcal{G}_u \quad (\text{A.6c})$$

$$1/M := A^{-1} \partial_T A|_{\theta} = 2 \partial_T \xi|_{\theta} = 4 \partial_{\pi}\xi|_{\theta} = -4\rho \partial_{\pi\pi}\mathcal{G}_u \quad (\text{A.6d})$$

where C is the specific heat at constant pressure, α is the areal coefficient of thermal expansion, and M is the modulus of dilation (uniform expansion). Here η is entropy, θ is temperature, $T := \frac{1}{2}(\sigma_{11} + \sigma_{22})$ is surface tension (an invariant in 2D whose intensive variable is $\pi = 2T = \sigma_{11} + \sigma_{22}$ wherein σ_{ij} are components of Cauchy stress quantified in 2D), and A is area (whose extensive variable is $\xi = \ln \sqrt{A/A_0}$, which denotes dilation, i.e., that strain describing an uniform areal expansion).

Appendix A.3.2. Non-Uniform Response

Because $d\mathcal{U} = d\mathcal{U}_u + d\mathcal{U}_n$ with $d\mathcal{U}_u = \theta d\eta + \frac{1}{\rho}\pi d\xi$ it follows that $d\mathcal{U}_n = \frac{1}{\rho}(\sigma d\varepsilon + \tau d\gamma)$ out of which come constitutive equations that govern the non-uniform response of a Green elastic membrane, viz., $\sigma = \rho \partial_{\varepsilon}\mathcal{U}_n$ and $\tau = \rho \partial_{\gamma}\mathcal{U}_n$ which, assuming these are continuous and differentiable functions of state, become the following system of differential equations

$$\begin{Bmatrix} d\sigma \\ d\tau \end{Bmatrix} = \rho \begin{bmatrix} \partial_{\varepsilon\varepsilon}\mathcal{U}_n & \partial_{\varepsilon\gamma}\mathcal{U}_n \\ \partial_{\gamma\varepsilon}\mathcal{U}_n & \partial_{\gamma\gamma}\mathcal{U}_n \end{bmatrix} \begin{Bmatrix} d\varepsilon \\ d\gamma \end{Bmatrix} = \begin{bmatrix} 2N & 2\tau \\ 2\tau & G \end{bmatrix} \begin{Bmatrix} d\varepsilon \\ d\gamma \end{Bmatrix} \quad (\text{A.7a})$$

with material constants defined accordingly

$$N := \Gamma \partial_{\Gamma}(\mathcal{S}_{11} - \mathcal{S}_{22})|_g = \frac{1}{2} \partial_{\varepsilon}\pi|_{\gamma} = \frac{1}{2}\rho \partial_{\varepsilon\varepsilon}\mathcal{U}_n \quad (\text{A.7b})$$

$$G := \Gamma \partial_g \mathcal{S}_{21}|_{\Gamma} = \partial_{\gamma}\tau|_{\varepsilon} = \rho \partial_{\gamma\gamma}\mathcal{U}_n \quad (\text{A.7c})$$

subject to the constraint

$$2\tau = \rho \partial_{\varepsilon\gamma} \mathcal{U}_n = \rho \partial_{\gamma\varepsilon} \mathcal{U}_n \quad (\text{A.7d})$$

which follows from Eqn. (4.14). This constraint suggests that the non-uniform response is actually Rajagopal elastic, because τ is a response variable, with \mathcal{U}_n therefore being an implicit function of state.

Appendix A.3.3. A Hookean Membrane

A Green elastic membrane whose uniform response is governed by a Gibbs free-energy function of the form

$$\mathcal{G}_u(\theta, \pi) = -C \left(\theta \ln \left(\frac{\theta}{\theta_0} \right) - (\theta - \theta_0) \right) - \frac{\pi}{2\rho} \left(\alpha(\theta - \theta_0) + \frac{\pi}{4M} \right) \quad (\text{A.8a})$$

and whose non-uniform response is governed by an internal energy function of the form

$$\mathcal{U}_n(\varepsilon, \gamma) = \mathcal{U}_n(-\varepsilon, \gamma) = \mathcal{U}_n(\varepsilon, -\gamma) = \mathcal{U}_n(-\varepsilon, -\gamma) = \frac{1}{\rho} (2\tau\varepsilon\gamma + N\varepsilon^2 + \frac{1}{2}G\gamma^2) \quad (\text{A.8b})$$

with negative strain arguments accounting for symmetries in strain, which are pertinent for all terms except for the coupling term $2\tau\varepsilon\gamma$. Form these formulæ come the following system of differential equations

$$\begin{Bmatrix} d\eta \\ d\pi \\ d\sigma \\ d\tau \end{Bmatrix} = \begin{bmatrix} C/\theta - M\alpha^2/\rho & 2M\alpha/\rho & 0 & 0 \\ -2M\alpha & 4M & 0 & 0 \\ 0 & 0 & 2N & 2\tau \\ 0 & 0 & 2\tau & G \end{bmatrix} \begin{Bmatrix} d\theta \\ d\xi \\ d\varepsilon \\ d\gamma \end{Bmatrix} \quad (\text{A.8c})$$

which describes a thermoelastic Hookean membrane whose material parameters ρ , C , α , M , N and G are all constant valued across state space.

Appendix A.4. Alveolar Septa as Rajagopal (Implicit) Thermoelastic Membranes

We employ implicit elasticity here to derive a constitutive theory suitable for describing biologic membranes.

Appendix A.4.1. Biologic Membrane Under Uniform Motions

Like the implicit elastic fiber introduced in Eqn. (A.5), the uniform response of an implicit elastic membrane with a strain-limiting dilation can be modeled using a Gibbs free energy $\mathcal{G}_u(\theta, \xi, \pi) := \mathcal{G}_1(\xi_1, \pi) + \mathcal{G}_2(\theta, \pi)$ where the dilation $\xi := \ln \sqrt{A/A_0}$ is considered to decompose into a sum of two dilations: $\xi_1 := \ln \sqrt{A_1/A_0}$ and $\xi_2 := \ln \sqrt{A/A_1}$ so that $\xi = \xi_1 + \xi_2$, with like interpretations as those from their linear counterparts, viz., e , e_1 and e_2 ; specifically,

$$C := \theta \partial_{\theta} \eta|_{\pi} = -\theta \partial_{\theta\theta} \mathcal{G}_2(\theta, \pi) \quad (\text{A.9a})$$

$$\alpha := \partial_{\theta} \xi|_{\pi} = -\rho \partial_{\pi\theta} \mathcal{G}_2(\theta, \pi) \equiv -\rho \partial_{\theta\pi} \mathcal{G}_2(\theta, \pi) \quad (\text{A.9b})$$

$$1/M := 4 \partial_{\pi} \xi|_{\theta} = -4 \left((\rho \partial_{\xi_1} \mathcal{G}_1(\xi_1, \pi))^{-1} (\xi + \rho \partial_{\pi} \mathcal{G}_u(\theta, \xi, \pi)) + \rho \partial_{\pi\pi} \mathcal{G}_2(\theta, \pi) \right) \quad (\text{A.9c})$$

whose derivation is analogous to that of the implicit fiber above. In this case, we consider a Gibbs free-energy function of the form

$$\mathcal{G}_1(\xi_1, \pi) = -\frac{1}{\rho} \left(\xi_t(4M_1\xi_1 - \pi) + 2\xi_1\pi \right) \quad (\text{A.10a})$$

$$\mathcal{G}_2(\theta, \pi) = -C \left(\theta \ln \left(\frac{\theta}{\theta_0} \right) - (\theta - \theta_0) \right) - \frac{\pi}{2\rho} \left(\alpha(\theta - \theta_0) + \frac{\pi}{4M_2} \right) \quad (\text{A.10b})$$

whose resulting elastic compliance is

$$\frac{1}{M(\theta, \xi, \pi)} = \frac{\xi_t - \xi_1}{M_1\xi_t + \pi/2} + \frac{1}{M_2} \quad \text{wherein} \quad \xi_1 = \xi - \frac{1}{2}\alpha(\theta - \theta_0) - \frac{\pi}{4M_2} \quad (\text{A.10c})$$

with $\xi_t > 0$ being an upper bound on strain ξ_1 in that $0 \leq \xi_1 < \xi_t$. Such a membrane has an initial tangent stiffness $M(\theta_0, 0, 0)$ of $M_1M_2/(M_1 + M_2) \approx M_1$ whenever $M_2 \gg M_1 > 0$, and a terminal tangent stiffness $M(\xi_1 = \xi_t)$ of M_2 .

Membranes will wrinkle under states of negative surface tension (or dilation). In alveolar mechanics, surfactant helps to prevent this, and a possible ensuing alveolar collapse. Wrinkling is not accounted for in our modeling of alveolar septa. Rather, like fibers, membranes are assumed to support compression with a modulus of $M_1M_2/(M_1 + M_2)$, which associates with the compliant response found at the origin (zero tension, zero dilation). This is done to help ensure numeric stability in our software.

The difference between a Green and Rajagopal thermoelastic membrane undergoing a dilation is in their definitions for elastic compliance. There is no difference in their properties for specific heat or thermal expansion. The above model has been successfully applied to a visceral pleura membrane [44].

Appendix A.5. Biologic Membrane Under Non-Uniform Motions

We seek an energetic construction that is consistent with that of the Freed & Rajagopal fiber model [68], but which is applicable to the non-uniform responses of planar membranes. A Rajagopal elastic solid is implicit; therefore, we consider an internal energy with the following special structure

$$\mathcal{U}_n(\varepsilon, \gamma, \sigma, \tau) = \frac{1}{\rho} 2\varepsilon\gamma\tau + \mathcal{U}_1(\varepsilon_1, \sigma) + \mathcal{U}_2(\varepsilon_2) + \mathcal{U}_3(\gamma_1, \tau) + \mathcal{U}_4(\gamma_2) \quad (\text{A.11})$$

which depends upon squeeze strains $\varepsilon := \ln \sqrt{\Gamma/\Gamma_0}$, $\varepsilon_1 := \ln \sqrt{\Gamma_1/\Gamma_0}$ and $\varepsilon_2 := \ln \sqrt{\Gamma/\Gamma_1}$, and shear strains $\gamma := g - g_0$, $\gamma_1 := g_1 - g_0$ and $\gamma_2 := g - g_1$, both of which are additive in that $\varepsilon = \varepsilon_1 + \varepsilon_2$ and $\gamma = \gamma_1 + \gamma_2$, and as such, so are their differential rates of change $d\varepsilon = d\varepsilon_1 + d\varepsilon_2$ and $d\gamma = d\gamma_1 + d\gamma_2$. Strains ε_1 and γ_1 may be thought of as describing unravelings of molecular configuration, analogous to e_1 in the fiber model of Eqn. (A.5) and ξ_1 in the uniform membrane model of Eqn. (A.10c). The first term on the right-hand side of Eqn. (A.11) ensures that the constraint in Eqn. (A.7d) is satisfied. Other than this term, no coupling between squeeze and shear is assumed in this energy function. Energies \mathcal{U}_1 and \mathcal{U}_3 are Rajagopal elastic (implicit), while energies \mathcal{U}_2 and \mathcal{U}_4 are Green elastic (explicit).

Given a non-uniform internal energy in the form of Eqn. (A.11), then the squeeze compliance is found to be

$$\frac{1}{N} = 2 \left(\frac{\rho \partial_{\sigma} \mathcal{U}_1(\varepsilon_1, \sigma)}{\rho \partial_{\varepsilon_2} \mathcal{U}_2(\varepsilon_2) - \rho \partial_{\varepsilon_1} \mathcal{U}_1(\varepsilon_1, \sigma)} + \frac{1}{\rho \partial_{\varepsilon_2 \varepsilon_2} \mathcal{U}_2(\varepsilon_2)} \right) \quad (\text{A.12a})$$

while the shear compliance is found to be

$$\frac{1}{G} = \Gamma \left(\frac{\rho \partial_{\tau} \mathcal{U}_3(\gamma_1, \tau) + 2\varepsilon\gamma}{\rho \partial_{\gamma_2} \mathcal{U}_4(\gamma_2) - \rho \partial_{\gamma_1} \mathcal{U}_3(\gamma_1, \tau)} + \frac{1 - 2\varepsilon}{\rho \partial_{\gamma_2 \gamma_2} \mathcal{U}_4(\gamma_2)} \right) \quad (\text{A.12b})$$

whose mathematical structure is similar to that of the fiber model presented in Eqn. (A.5). The first terms in the parentheses are Rajagopal elastic. The second terms are Green elastic. The shear compliance $1/G$ has extra terms of $2\varepsilon\gamma$ and $(1 - 2\varepsilon)$ that arise because of the constraint energy $\frac{1}{\rho} 2\varepsilon\gamma\tau$, which is a direct consequence of defining shear as $\tau := \Gamma \mathcal{S}_{21}$. These extra terms are missing in the original, implicit, membrane model derived by Freed *et al.* [44].

Derivation: The First and Second Laws of Thermodynamics, as they pertain to non-uniform contributions of stress power, obey $d\mathcal{U}_n(\varepsilon, \gamma, \sigma, \tau) = \frac{1}{\rho} dW_n$ where, from Eqn. (A.11), one gets $\rho d\mathcal{U}_n = (2\gamma\tau + \rho \partial_{\varepsilon_1} \mathcal{U}_1(\varepsilon_1, \sigma))d\varepsilon_1 + (2\gamma\tau + \rho \partial_{\varepsilon_2} \mathcal{U}_2(\varepsilon_2))d\varepsilon_2 + (\rho \partial_{\sigma} \mathcal{U}_1(\varepsilon_1, \sigma))d\sigma + (2\varepsilon\tau + \rho \partial_{\gamma_1} \mathcal{U}_3(\gamma_1, \tau))d\gamma_1 + (2\varepsilon\tau + \rho \partial_{\gamma_2} \mathcal{U}_4(\gamma_2))d\gamma_2 + (2\varepsilon\gamma + \rho \partial_{\tau} \mathcal{U}_3(\gamma_1, \tau))d\tau$ while the right side becomes $dW_n = \sigma d\varepsilon_1 + \sigma d\varepsilon_2 + \tau d\gamma_1 + \tau d\gamma_2$ because of the additivity in strain rates. Gathering like terms result in two Green elastic formulæ that produce the stresses

$$\sigma = 2\gamma\tau + \rho \partial_{\varepsilon_2} \mathcal{U}_2(\varepsilon_2) \quad \text{and} \quad \tau = 2\varepsilon\tau + \rho \partial_{\gamma_2} \mathcal{U}_4(\gamma_2)$$

and two Rajagopal elastic formulæ that govern the internal strains

$$\begin{aligned} \rho \partial_{\sigma} \mathcal{U}_1(\varepsilon_1, \sigma) d\sigma &= (\pi - 2\gamma\tau - \rho \partial_{\varepsilon_1} \mathcal{U}_1(\varepsilon_1, \sigma))d\varepsilon_1 \\ (2\varepsilon\gamma + \rho \partial_{\tau} \mathcal{U}_3(\gamma_1, \tau))d\tau &= (\tau(1 - 2\varepsilon) - \rho \partial_{\gamma_1} \mathcal{U}_3(\gamma_1, \tau))d\gamma_1 \end{aligned}$$

that when combined produce the constitutive responses

$$\begin{aligned} \rho \partial_{\sigma} \mathcal{U}_1(\varepsilon_1, \sigma) d\sigma &= (\rho \partial_{\varepsilon_2} \mathcal{U}_2(\varepsilon_2) - \rho \partial_{\varepsilon_1} \mathcal{U}_1(\varepsilon_1, \sigma))d\varepsilon_1 \\ (2\varepsilon\gamma + \rho \partial_{\tau} \mathcal{U}_3(\gamma_1, \tau))d\tau &= (\rho \partial_{\gamma_2} \mathcal{U}_4(\gamma_2) - \rho \partial_{\gamma_1} \mathcal{U}_3(\gamma_1, \tau))d\gamma_1. \end{aligned}$$

We now differentiate the Green elastic constitutive equations, thereby putting them into differential form. When doing this, we impose a constraint that the squeeze compliance $1/N$ is to be evaluated at constant shear γ , while the shear compliance $1/G$ is to be evaluated at constant squeeze ε , consequently

$$d\sigma|_{\gamma} = \rho \partial_{\varepsilon_2 \varepsilon_2} \mathcal{U}_2(\varepsilon_2) d\varepsilon_2 \quad \text{and} \quad d\tau|_{\varepsilon} = 2\varepsilon d\tau + \rho \partial_{\gamma_2 \gamma_2} \mathcal{U}_4(\gamma_2) d\gamma_2.$$

Because $d\varepsilon = d\varepsilon_1 + d\varepsilon_2$ and $d\gamma = d\gamma_1 + d\gamma_2$, one solves the above equations for strain rate, adds them appropriately, and from these one can thereby construct the two compliances found in Eqn. (A.12) via the expressions

$$\frac{1}{N} = 2 \left. \frac{d\varepsilon}{d\sigma} \right|_{\gamma} = 2 \left(\left. \frac{d\varepsilon_1}{d\sigma} \right|_{\gamma} + \left. \frac{d\varepsilon_2}{d\sigma} \right|_{\gamma} \right) \quad \text{and} \quad \frac{1}{G} = \Gamma \left. \frac{d\gamma}{d\tau} \right|_{\varepsilon} = \Gamma \left(\left. \frac{d\gamma_1}{d\tau} \right|_{\varepsilon} + \left. \frac{d\gamma_2}{d\tau} \right|_{\varepsilon} \right)$$

that when all terms are collected together become Eqn. (A.12). \square

We now seek an energy function (A.11) that produces compliances $1/N$ and $1/G$ with a like mathematical structure to that of Eqn. (A.10c) for dilation, viz., $1/M$; specifically, we shall consider

$$\rho\mathcal{U}_1(\varepsilon_1, \sigma) = \varepsilon_t(N_1\varepsilon_1 - \sigma) + \varepsilon_1\sigma \quad \rho\mathcal{U}_2(\varepsilon_2) = \frac{1}{2}N_2\varepsilon_2^2 \quad (\text{A.13a})$$

$$\rho\mathcal{U}_3(\gamma_1, \tau) = \gamma_t(G_1\gamma_1 - \tau) + \gamma_1\tau \quad \rho\mathcal{U}_4(\gamma_2) = \frac{1}{2}G_2\gamma_2^2 \quad (\text{A.13b})$$

which have the same mathematical structure as the energies for the biologic fiber (A.5) and uniform membrane (A.10). When substituted into Eqn. (A.12), they produce the following thermoelastic compliances

$$\frac{1}{N} = 2 \left(\frac{\varepsilon_t - |\varepsilon_1|}{N_1\varepsilon_t + 2\gamma\tau} + \frac{1}{N_2} \right) \quad \varepsilon_1 = \varepsilon - \frac{\sigma - 2\gamma\tau}{N_2} \quad (\text{A.14a})$$

$$\frac{1}{G} = \Gamma \left(\frac{\gamma_t - 2\varepsilon\gamma - |\gamma_1|}{G_1\gamma_t + 2\varepsilon\tau} + \frac{1 - 2\varepsilon}{G_2} \right) \quad \gamma_1 = \gamma - \frac{(1 - 2\varepsilon)\tau}{G_2} \quad (\text{A.14b})$$

where absolute values are introduced because the squeeze ε and shear γ strains can take on both positive and negative values, viz., they are odd functions, cf. [44]. (Recall that in our construction the material parameters are tangents to response curves—the models are differential.) Like our other biologic models, the squeeze compliance $1/N$ is described by three material parameters: an asymptotic modulus at the reference state of $N_1N_2/(N_1 + N_2) \approx N_1$ whenever $N_2 \gg N_1 > 0$, where N_1 may be thought of as the stiffness of an unstretched molecular network; a terminal modulus N_2 designating a stiffness after this molecular network has been stretched out; and a limiting state of configurational squeeze ε_t . The shear compliance $1/G$ is also described by three material parameters: an asymptotic modulus at the reference state of $G_1G_2/(G_1 + G_2) \approx G_1$ whenever $G_2 \gg G_1 > 0$, a terminal modulus G_2 , and a limiting state of configurational shear γ_t that shifts by $2\varepsilon\gamma$. This shift is a consequence of the stress-strain coupling introduced in shear $\tau := \Gamma\mathcal{S}_{21}$.

Appendix B. Overview

These appendices describe interfaces for a software package written in Python whose intent is to model the micro-mechanical response of alveolar sacs that comprise the bulk of the parenchyma in lung tissue. A flow chart for this software is presented in Fig. B.1. The software assumes the following design strategy: *i*) a reference configuration exists (see ¶ below) and three sequential configurations exist that are separated in time by an uniform time step, *ii*) the three sequential configurations associate with steps $n-1$, n and $n+1$, *iii*) co-ordinates for the next configuration are assigned through a vertex's **update** method (they can be reassigned multiple times at any step along a solution path), *iv*) the **advance** method relabels the current data to their associated previous data, and then relabels the next data to their associated current data, thereby preparing the data structure of each object within a dodecahedral object for its next step along a solution path, and *v*) the mechanical response is isotropic and can be described by three modes of deformation: dilatation/dilation, squeeze and shear [44, 45].

The initial co-ordinates that locate each vertex in a dodecahedron used to model the alveoli of lung are assigned according to a reference configuration where the pleural pressure (the negative pressure surrounding lung in the pleural cavity) and the transpulmonary pressure (the difference between aleolar and pleural pressures) are both at zero gauge pressure, i.e., *all pressures are atmospheric pressure in the reference state*. This is not a typical physiological state. The pleural pressure is normally negative, sucking the pleural membrane against the wall of the chest. During expiration, the diaphragm is pushed up, reducing the pleural pressure. The pleural pressure remains negative during breating at rest, but it can become positive during active expiration. The surface tension created by surfactant helps keep most alveoli open during excursions into positive pleural pressures, but not all will remain open. Alveolar recruitment is not addressed here. Under normal conditions, alveoli are their smallest at max expiration. Alveolar size is predominately determined by the transpulmonary pressure. The greater the transpulmonary pressure the greater the alveolar size.

Numerous methods belonging to the classes of these appendices have a string argument that is denoted as **state** which can take on any of the following values:

'c', 'curr', 'current' gets the value for a current configuration

'n', 'next' gets the value for a next configuration

'p', 'prev', 'previous' gets the value for a previous configuration

'r', 'ref', 'reference' gets the value for the reference configuration

Several strings can be used to denote each **state**.

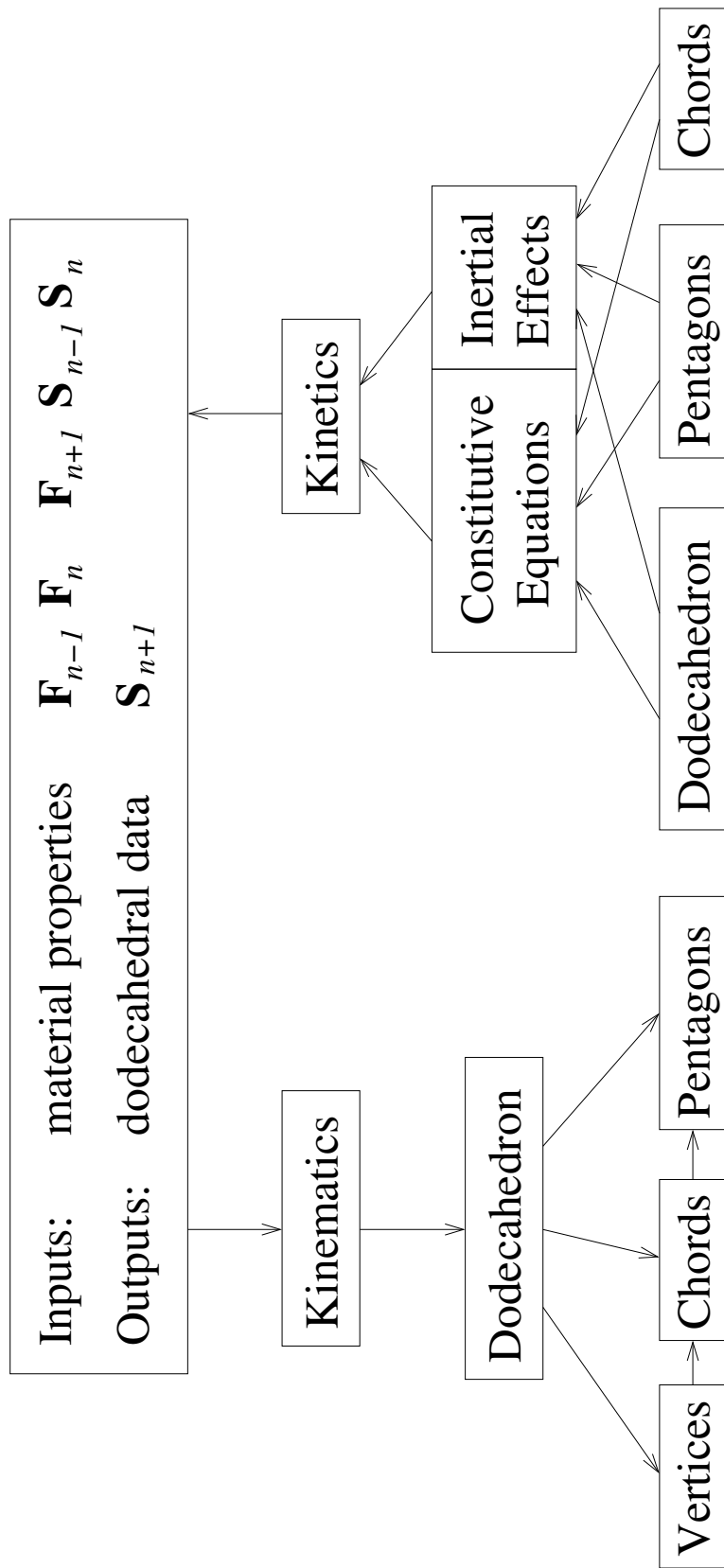


Figure B.1: Flow chart for a dodecahedral model of an alveolus.

Appendix C. Dodecahedra

Module `dodecahedra.py` is Python code that exports class `dodecahedron`. An object of type `dodecahedron` is comprised of twenty vertices labeled according to Table 2.1, as visualized in Fig. 2.1, thirty chords assigned according to Table 2.2, and twelve pentagons assigned according to Table 2.3. This class has the following interface:

class `dodecahedron`

constructor

```
d = dodecahedron(F0, h, gaussPts=1, alveolarDiameter=1.952400802898434)
```

`F0` a deformation gradient: distortion from regular shape in a reference state
`h` time step separating two neighboring configurations
`gaussPts` number of Gauss points in each pentagonal surface: $\in \{1, 4, 7\}$
`alveolarDiameter` mean diameter of an alveolar sac

The default alveolar diameter results in vertices of the dodecahedron taking on co-ordinate values that associate with its natural configuration, i.e., all vertices touch the unit sphere from within. Adopting the labeling scheme presented in Fig. 2.1, the vertices are indexed according to Table 2.1, the chords are indexed according to Table 2.2, and the pentagons are indexed according to Table 2.3. If `F0` is the identity matrix, then the shape will be that of a regular dodecahedron in its reference state; otherwise, the shape will be that of an irregular dodecahedron in its reference state. The number of `gaussPts`, viz., 1, 4 or 7, establishes the quadrature scheme to be used for numeric integration, in accordance with Fig. 5.1.

methods

```
s = d.verticesToString(state)
```

Returns a formatted string description for this dodecahedron's vertices in configuration `state`.

```
s = d.chordsToString(state)
```

Returns a formatted string description for this dodecahedron's chords in configuration `state`.

```
s = d.pentagonsToString(state)
```

Returns a formatted string description for this dodecahedron's pentagons in configuration `state`.

```
v = d.getVertex(number)
```

Returns that vertex indexed with `number`, which must be in interval $[1, 20]$.

```
c = d.getChord(number)
```

Returns that chord indexed with `number`, which must be in interval $[1, 30]$.

```
p = d.getPentagon(number)
```

Returns that irregular pentagon indexed with `number`, which must be in interval $[1, 12]$.

d.update(nextF)

Assuming that the deformation imposed on an alveolus is homogeneous, described by a deformation gradient '**nextF**', this procedure assigns new co-ordinate values to all vertices of the dodecahedron for its next configuration such that whenever **nextF** is the identity matrix the dodecahedron is in its reference state. This method calls the **update** methods for all of its vertices, chords and pentagons, after which it updates the local fields of the dodecahedron object itself. This method may be called multiple times before freezing its values with a call to **advance**.

d.advance()

Calls method **advance** for all of the vertices, chords and pentagons comprising this dodecahedron, where current fields are assigned to previous fields, and then next fields are assigned to current fields for these objects. Afterwards, it assigns the current fields to the previous fields and then assigns the next fields to the current fields of the dodecahedron itself, thereby freezing the present next-fields in preparation for advancing the solution along its path.

The geometric fields associated with a dodecahedron.

v = d.volume(state)

Returns the volume of this dodecahedron in configuration **state**.

vLambda = d.volumetricStretch(state)

Returns the cube root of the volume at **state** divided by reference volume, i.e., $\sqrt[3]{V/V_0}$.

vStrain = d.volumetricStrain(state)

Returns the logarithm of volumetric stretch evaluated at **state**, i.e., $\Xi = \ln \sqrt[3]{V/V_0}$.

dvStrain = d.dVolumetricStrain(state)

Returns the rate of volumetric strain at **state**, viz., $d\Xi = \frac{1}{3}V^{-1}dV$.

Appendix D. Vertices

Module `vertices.py` is Python code that exports class `vertex`. There are twenty vertices in a dodecahedron. Their normalized reference co-ordinates are presented in Table 2.1, which are indexed according to Fig. 2.1. These normalized co-ordinates are uniformly scaled by the factor `alveolarDiameter/1.952400802898434` supplied to the `dodecahedron` constructor, and then transformed by the linear operator `F0` also supplied to the `dodecahedron` constructor; the vertices are created within the `dodecahedron` constructor. (The user does not call the `vertex` constructor.) This module has the following interface:

function

```
s = coordinatesToString(x, y, z)
    x  the 1 co-ordinate
    y  the 2 co-ordiante
    z  the 3 co-ordiante
```

Returns a formatted string representation for the assigned set of co-ordinates.

class vertex

constructor

```
v = vertex(number, x0, y0, z0, h)
    number  an immutable value unique to this vertex
    x0      the initial x co-ordinate at zero pleural pressure
    y0      the initial y co-ordinate at zero pleural pressure
    z0      the initial z co-ordinate at zero pleural pressure
    h       the time-step size between two neighboring configurations
```

co-ordinates `x0`, `y0`, `z0` have values assigned in the reference co-ordinate frame of a dodecahedron. The natural co-ordinates for the vertices of a regular dodecahedron are listed in Table 2.1.

methods

```
s = v.toString(state)
```

Returns a formatted string representation for this vertex in configuration `state` of its dodecahedron.

```
n = v.number()
```

Returns the unique number affiliated with this vertex.

```
x, y, z = v.coordinates(state)
```

Returns the co-ordinates for this vertex in configuration `state`, which are evaluated in the co-ordinate system of its dodecahedron.

`v.update(x, y, z)`

Assigns a new set of co-ordinate values to the vertex affiliated with the next configuration of its dodecahedron, as quantified in the co-ordinate system of its dodecahedron. This method may be called multiple times before freezing its value with a call to `advance`. (This method is called internally by `dodecahedron` objects.)

`v.advance()`

Assigns all of the object's data associated with the current configuration into their affiliated data associated with the previous configuration, and then assigns all of the object's data associated with the next configuration into their affiliated data associated with the current configuration, thereby freezing these data from external change. (This method is called internally by `dodecahedron` objects.)

Kinematic fields associated with a point (vertex) in 3 space.

`[ux, uy, uz] = v.displacement(state)`

Returns the displacement vector of this vertex for configuration `state` whose components are evaluated in the co-ordinate system of its dodecahedron. Displacements interpolate quadratically between consecutive states, because only three locations are maintained at any step n along a solution path.

`[vx, vy, vz] = v.velocity(state)`

Returns the velocity vector of this vertex for configuration `state` whose components are evaluated in the co-ordinate system of its dodecahedron. Velocities are calculated using second-order difference formulæ. Velocities interpolate linearly between consecutive states, because only three locations are maintained at any step n along a solution path.

`[ax, ay, az] = v.acceleration(state)`

Returns the acceleration vector of this vertex for configuration `state` whose components are evaluated in the co-ordinate system of its dodecahedron. Accelerations are equivalent for the previous, current and next states, i.e., accelerations are constant over an interval $(n-1, n+1)$; consequently, accelerations are discontinuous along a solution path. This is because only three locations are maintained at any step n along a solution path.

Appendix E. Chords

Module `chords.py` is Python code that exports class `chord`. There are thirty chords in a dodecahedron. They are assigned vertices according to Table 2.2 that index according to Fig. 2.1. They are created within the `dodecahedron` constructor. (The user does not call the `chord` constructor.) This class has the following interface:

class `chord`

constructor

```
c = chord(number, vertex1, vertex2, h)
    number    an immutable value unique to this chord
    vertex1   an end point of the chord, an object of class vertex
    vertex2   an end point of the chord, an object of class vertex
    h         timestep size between two neighboring configurations
```

Vertices `vertex1` and `vertex2` must be different. The chordal numbering scheme is specified in Table 2.2, given the vertex numbering scheme for the dodecahedron listed in Table 2.1 that is visible in Fig. 2.1.

methods

```
s = c.toString(state)
```

Returns a formatted string representation for this chord in configuration `state` of its dodecahedron.

```
n = c.number()
```

Returns the unique number affiliated with this chord.

```
v1, v2 = c.vertexNumbers()
```

Returns the vertex numbers assigned to the two vertices of this chord.

```
truth = c.hasVertex(number)
```

Returns `True` if one of the two vertices has this vertex number; otherwise, it returns `False`.

```
v = c.getVertex(number)
```

Returns the vertex with identifier `number`. Typically, it is to be called inside a `c.hasVertex` `if` clause.

```
c.update()
```

Establishes the fields that pertain to this instance of `chord` which affiliate with the next configuration. It is to be called after all vertices have had their co-ordinates updated. This method does **not** call the `update` method for the two vertices at its end points. This method may be called multiple times before freezing its values with a call to `advance`. (This method is called internally by `dodecahedron` objects.)

`c.advance()`

Assigns all of the object's data associated with the current configuration into their affiliated data associated with the previous configuration, and then assigns all of the object's data associated with the next configuration into their affiliated data associated with the current configuration, thereby freezing these data from external change. This method does **not** call the `advance` method for the two vertices at its end points. (This method is called internally by `dodecahedron` objects.)

The geometric fields associated with a chord in 3 space.

`ell = c.length(state)`

Returns the chordal length in configuration `state` of its dodecahedron.

`lambda = c.stretch(state)`

Returns the stretch of this chord in configuration `state` of its dodecahedron.

The kinematic fields associated with the centroid of a chord in 3 space.

`[x, y, z] = c.centroid(state)`

Returns the position vector for this chord locating its mid-point in configuration `state` of its dodecahedron, i.e., it is the χ vector of Fig. 2.4.

`[ux, uy, uz] = c.displacement(state)`

Returns the displacement vector of the centroid for this chord in configuration `state` of its dodecahedron.

`[vx, vy, vz] = c.velocity(state)`

Returns the velocity vector of the centroid for this chord in configuration `state` of its dodecahedron.

`[ax, ay, az] = c.acceleration(state)`

Returns the acceleration vector of the centroid for this chord in configuration `state` of its dodecahedron.

The rotation and spin matrices for this chord, as measured relative to its dodecahedron's co-ordinate system.

`pMtx = c.rotation(state)`

Returns a 3×3 orthogonal matrix \mathbf{P} that rotates the base vectors from its dodecahedral frame of reference into a set of local base vectors where the 1 direction is tangent to the chordal axis, the 2 direction is the normal for this curve in 3 space, and the 3 direction is its binormal. The returned matrix associates with configuration `state` of its dodecahedron.

`omegaMtx = c.spin(state)`

Returns a 3×3 skew symmetric matrix $\mathbf{\Omega} := \dot{\mathbf{P}}\mathbf{P}^T$ that describes the time rate of rotation, i.e., the spin of the local chordal co-ordinate system about the fixed system of its dodecahedron. The returned matrix associates with configuration `state` of its dodecahedron.

The thermodynamic strain and strain-rate fields associated with a chord.

`strain = c.strain(state)`

Returns the logarithmic strain for this chord in configuration **state** of its dodecahedron, i.e., $e = \ln(L/L_0)$.

`dStrain = c.dStrain(state)`

Returns the logarithmic strain-rate for this chord in configuration **state** of its dodecahedron, viz., $de = L^{-1} dL$.

Appendix F. Modules for Planar Pentagons

There are three separate modules that collectively allow a description to be cast for a response of the twelve pentagons that comprise the surface of a dodecahedron. The first module provides their shape functions. The second module provides their kinematics, modeled as membranes. While the third module incorporates these features into a viable description for pentagons.

Appendix F.1. Shape Functions

Module **shapeFunctions** is Python code that exports class **shapeFunction**. These shape functions are for the geometry of an irregular planar pentagon. Shape functions for other geometries are not needed. These objects are created within the **pentagon** constructor, and are utilized by the objects of that class. Class **shapeFunction** has the following interface:

class shapeFunction

constructor

```
sf = shapeFunction(xi, eta)
    xi    the x co-ordinate in the natural co-ordinate system
    eta   the y co-ordinate in the natural co-ordinate system
```

where admissible co-ordinates **xi** and **eta** are any values that lie within the area of that pentagon which inscribes a unit circle or that reside along its boundary, as drawn in Fig. 2.2.

methods

```
y = sf.interpolate(y1, y2, y3, y4, y5)
```

Returns an interpolation for field **y** at location (**xi**, **eta**) given values **y1**, **y2**, **y3**, **y4**, **y5** for some field of interest, which are evaluated at the five vertices of the pentagon, indexed according to Fig. 2.2. Arguments may be of any numeric type or that of a NumPy array.

```
Gmtx = sf.G(x1, x2, x3, x4, x5, x01, x02, x03, x04, x05)
```

Returns the displacement gradient

$$\mathbf{G} = \begin{bmatrix} \partial u / \partial x & \partial u / \partial y \\ \partial v / \partial x & \partial v / \partial y \end{bmatrix} \quad \text{where} \quad \begin{aligned} u &= x - X \\ v &= y - Y \end{aligned}$$

at that location with natural co-ordinates (**xi**, **eta**) residing within a pentagonal plane. The arguments are tuples providing co-ordinate positions for the vertices of a pentagon in their pentagonal frame of reference, e.g., **x1** = (x_1, y_1), ..., **x01** = (X_1, Y_1), ... with subscripts indexing according to Fig. 2.2.

```
Fmtx = sf.F(x1, x2, x3, x4, x5, x01, x02, x03, x04, x05)
```

Returns the deformation gradient

$$\mathbf{F} = \begin{bmatrix} \partial x / \partial X & \partial x / \partial Y \\ \partial y / \partial X & \partial y / \partial Y \end{bmatrix}$$

at that location with natural co-ordinates (**xi**, **eta**) residing within a pentagonal plane. The arguments are tuples providing co-ordinate positions for the vertices of a pentagon in their pentagonal frame of reference, e.g., **x1** = (x_1, y_1), ..., **x01** = (X_1, Y_1), ... with subscripts indexing according to Fig. 2.2.

dFdXi = **sf.dFdXi**(**x1**, **x2**, **x3**, **x4**, **x5**, **x01**, **x02**, **x03**, **x04**, **x05**)

Returns the gradient of a deformation gradient taken with respect to the ξ direction

$$\frac{\partial \mathbf{F}}{\partial \xi} = \frac{\partial}{\partial \xi} \begin{bmatrix} \partial x / \partial X & \partial x / \partial Y \\ \partial y / \partial X & \partial y / \partial Y \end{bmatrix}$$

at that location with natural co-ordinates (**xi**, **eta**) residing within a pentagonal plane. The arguments are tuples providing co-ordinate positions for the vertices of a pentagon in their pentagonal frame of reference, e.g., **x1** = (x_1, y_1), ..., **x01** = (X_1, Y_1), ... with subscripts indexing according to Fig. 2.2.

dFdEta = **sf.dFdEta**(**x1**, **x2**, **x3**, **x4**, **x5**, **x01**, **x02**, **x03**, **x04**, **x05**)

Returns the gradient of a deformation gradient taken with respect to the η direction

$$\frac{\partial \mathbf{F}}{\partial \eta} = \frac{\partial}{\partial \eta} \begin{bmatrix} \partial x / \partial X & \partial x / \partial Y \\ \partial y / \partial X & \partial y / \partial Y \end{bmatrix}$$

at that location with natural co-ordinates (**xi**, **eta**) residing within a pentagonal plane. The arguments are tuples providing co-ordinate positions for the vertices of a pentagon in their pentagonal frame of reference, e.g., **x1** = (x_1, y_1), ..., **x01** = (X_1, Y_1), ... with subscripts indexing according to Fig. 2.2.

Appendix F.2. Membranes

Module **membranes.py** is Python code that exports class **membrane**. Objects of this class are used to describe kinematic fields at the Gauss points of a pentagon. These objects are created within the **pentagon** constructor, and are utilized by the objects of that class. Class **membrane** has the following interface:

class membrane

constructor

m = **membrane**(**h**)

h uniform time step separating any two neighboring configurations

methods

m.update(**nextF**)

Establishes the fields that pertain to this instance of **membrane** which affiliate with the deformation gradient **nextF** of the next configuration. This is a 2×2 matrix describing the deformation gradient in the plane of the pentagon. (It is not the deformation gradient sent to **d.update**(**nextF**) of **dodecahedron** objects.) This method may be called multiple

times before freezing its values with a call to `advance`. (This method is called internally by `pentagon` objects.)

`m.advance()`

Assigns all of the object's data associated with the current configuration into their affiliated data associated with the previous configuration, and then assigns all of the object's data associated with the next configuration into their affiliated data associated with the current configuration, thereby freezing these data from external change. (This method is called internally by `pentagon` objects.)

Tensor fields that associate with a Gram-Schmidt factorization of the deformation gradient.

`qMtx = m.Q(state)`

Returns the 2×2 re-indexing matrix that is applied to the deformation gradient prior to its Gram-Schmidt decomposition in configuration `state`.

`rMtx = m.R(state)`

Returns the 2×2 rotation matrix **Q** derived from a **QR** decomposition of the re-indexed deformation gradient in configuration `state`.

`omega = m.spin(state)`

Returns the 2×2 spin matrix caused by planar deformation, i.e., $d\mathbf{R}\mathbf{R}^\top$, in configuration `state`.

`uMtx = m.U(state)`

Returns the 2×2 Laplace stretch **R** (denoted herein as **U**) derived from a **QR** decomposition of the re-indexed deformation gradient in configuration `state`.

`uInvMtx = m.UInv(state)`

Returns the 2×2 inverse Laplace stretch **R** (denoted herein as **U**⁻¹) derived from a **QR** decomposition of the re-indexed deformation gradient in configuration `state`.

`duMtx = m.dU(state)`

Returns the differential change of the Laplace stretch **R** derived from a **QR** decomposition of the re-indexed deformation gradient in configuration `state`.

`duInvMtx = m.dUInv(state)`

Returns the differential change of the inverse Laplace stretch **R** derived from a **QR** decomposition of the re-indexed deformation gradient in configuration `state`.

Scalar attributes that arise as extensive thermodynamic variables.

`xi = m.dilation(state)`

Returns the planar dilation derived from a **QR** decomposition of the re-indexed deformation gradient in configuration `state`, i.e., $\xi = \ln \sqrt{A/A_0}$.

```
epsilon = m.squeeze(state)
```

Returns the in-plane squeeze derived from a **QR** decomposition of the re-indexed deformation gradient in configuration **state**, i.e., $\varepsilon = \ln \sqrt{\Gamma/\Gamma_0}$.

```
gamma = m.shear(state)
```

Returns the in-plane shear derived from a **QR** decomposition of the re-indexed deformation gradient in configuration **state**.

```
dXi = m.dDilation(state)
```

Returns a differential change in the planar dilation derived from a **QR** decomposition of the re-indexed deformation gradient in configuration **state**, viz., $d\xi = \frac{1}{2}A^{-1}dA$.

```
dEpsilon = m.dSqueeze(state)
```

Returns a differential change in the planar squeeze derived from a **QR** decomposition of the re-indexed deformation gradient in configuration **state**, viz., $d\varepsilon = \frac{1}{2}\Gamma^{-1}d\Gamma$.

```
dGamma = m.dShear(state)
```

Returns a differential change in the in-plane shear derived from a **QR** decomposition of the re-indexed deformation gradient in configuration **state**.

Appendix F.3. Pentagons

Module `pentagons.py` is Python code that exports class `pentagon`. There are twelve pentagons in a dodecahedron. They are assigned chords according to Table 2.3 with vertices assigned according to Table 2.1 that index according to Fig. 2.1. Pentagon objects are created within the `dodecahedron` constructor. Membrane and shape function objects are created within the `pentagon` constructor. Class `pentagon` has the following interface:

```
class pentagon
```

constructor

```
p = pentagon(number, chord1, chord2, chord3, chord4, chord5, h, gaussPts)
```

number	an immutable value unique to this pentagon
chord1	an edge of the pentagon, an object of class <code>chord</code>
chord2	an edge of the pentagon, an object of class <code>chord</code>
chord3	an edge of the pentagon, an object of class <code>chord</code>
chord4	an edge of the pentagon, an object of class <code>chord</code>
chord5	an edge of the pentagon, an object of class <code>chord</code>
h	timestep size between two neighboring configurations
gaussPts	number of Gauss points in a pentagonal surface: $\in \{1, 4, 7\}$

Chords `chord1`, `chord2`, `chord3`, `chord4`, and `chord5` must have five vertices that are common, assigned according to the scheme presented in Fig. F.2. The pentagon numbering scheme is specified in Table 2.3, given the vertex numbering scheme for the dodecahedron listed in Table 2.1. When assigning the five chords of a pentagon, do so according to

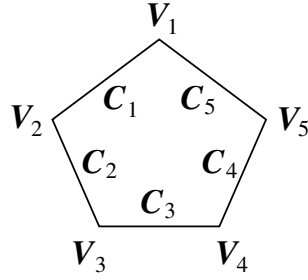


Figure F.2: Vertex and chord labeling scheme for a pentagon, which coincides with the labeling scheme used by the pentagonal shape functions.

Fig. F.2 when looking inward from the outside of a dodecahedron. By numbering the chords in a counterclockwise direction, the algorithm used to compute its area will be positive; otherwise, if the chords were numbered clockwise then the area derived by Eq. (2.9) would become negative.

methods

`s = p.toString(state)`

Returns a formatted string representation for this pentagon in configuration `state`.

`n = p.number()`

Returns the unique number affiliated with this pentagon.

`n1, n2, n3, n4, n5 = p.chordNumbers()`

Returns the chordal numbers associated with the chords of this pentagon, sorted from smallest to largest, i.e., not in accordance to Fig. F.2.

`n1, n2, n3, n4, n5 = p.vertexNumbers()`

Returns the vertex numbers associated with the vertices of this pentagon, sorted from smallest to largest, i.e., not in accordance to Fig. F.2.

`truth = p.hasChord(number)`

Returns `True` if one of the five chords has this chordal number; otherwise, it returns `False`.

`truth = p.hasVertex(number)`

Returns `True` if one of the five vertices has this vertex number; otherwise, it returns `False`.

`c = p.getChord(number)`

Returns the chord with specified `number`. Typically called inside a `p.haschord` if clause.

`v = p.getVertex(number)`

Returns the vertex with specified `number`. Typically called inside a `p.hasVertex` if clause.

`v = p.gaussPoints()`

Returns the number of Gauss points associated with this pentagon.

`p.update()`

Establishes the fields that pertain to this object of class **pentagon** that affiliate with the next configuration. It is to be called after all vertices and all chords have been updated. This method does **not** call the **update** methods for the five vertices nor for the five chords that comprise this pentagon, but it does call the **update** methods for the **shapeFunction** and **membrane** objects contained within. This method may be called multiple times before freezing its values with a call to **advance**. (This method is called internally by **dodecahedron** objects.)

`p.advance()`

Assigns all of the object's data associated with the current configuration into their affiliated data associated with the previous configuration, and then assigns all of the object's data associated with the next configuration into their affiliated data associated with the current configuration, thereby freezing these data from external change. This method does **not** call the **advance** methods for the five vertices nor for the five chords that make up this pentagon, but it does call the **advance** methods for the **shapeFunction** and **membrane** objects contained within. (This method is called internally by **dodecahedron** objects.)

The geometric fields associated with a pentagonal surface embedded in 3 space.

`a = p.area(state)`

Returns the area of this irregular pentagon in configuration **state**.

`aLambda = p.arealStretch(state)`

Returns the square root of the area at **state** divided by reference area, i.e., $\sqrt{A/A_0}$.

`aStrain = p.arealStrain(state)`

Returns the logarithm of areal stretch evaluated at 'state', i.e., $\ln \sqrt{A/A_0}$.

`daStrain = p.dArealStrain(state)`

Returns the rate of areal strain at 'state', viz., $\frac{1}{2}A^{-1}dA$.

`[nx, ny, nz] = p.normal(state)`

Returns the outward unit normal vector to this pentagon in configuration **state**.

`[dnx, dny, dnz] = p.dNormal(state)`

Returns the rate-of-change of the outward unit normal vector to this pentagon in configuration **state**.

Kinematic vector fields associated with the centroid of a pentagon in 3 space.

`[cx, cy, cz] = p.centroid(state)`

Returns the centroid of this irregular pentagon in configuration `state`, i.e., this is vector χ in Fig. 2.5.

`[ux, uy, uz] = p.displacement(state)`

Returns the displacement vector of its centroid in configuration `state`.

`[vx, vy, vz] = p.velocity(state)`

Returns the velocity vector of its centroid in configuration `state`.

`[ax, ay, az] = p.acceleration(state)`

Returns the acceleration vector of its centroid in configuration `state`.

The rotation and spin of a pentagonal surface as it moves through 3 space.

`pMtx = p.rotation(state)`

Returns a 3×3 orthogonal matrix \mathbf{P} that rotates the base vectors from its dodecahedral frame of reference into a set of local base vectors pertaining to this irregular pentagon whose outward normal aligns with the 3 direction, i.e., the irregular pentagon resides in the local pentagonal 1-2 plane. The local 1 direction connects the shoulders of the pentagon, while the 2 direction is rooted at the head of the pentagon. The returned matrix associates with configuration `state`.

`omegaMtx = p.spin(state)`

Returns a 3×3 skew symmetric matrix $\mathbf{\Omega} := d\mathbf{P}\mathbf{P}^T$ that describes the time rate of rotation, i.e., a spin of the local pentagonal co-ordinate system about the fixed co-ordinate system of its dodecahedron. The returned matrix associates with configuration `state`.

The kinematic tensor fields of a planar membrane evaluated in the reference co-ordinate system of the pentagon. To rotate a tensor field from the pentagonal frame, say $\bar{\mathbf{A}}$, into its dodecahedral frame producing \mathbf{A} , apply the following map: $\mathbf{A} = \mathbf{P}\bar{\mathbf{A}}\mathbf{P}^T$ where \mathbf{P} is the orthogonal matrix returned by `rotation`.

`fMtx = p.F(gaussPt, state)`

Returns the 2×2 planar deformation gradient \mathbf{F} located at `gaussPt` in configuration `state`.

`dFdX = p.dFdX(gaussPt, state)`

Returns the partial derivative taken with respect to the X direction of a 2×2 planar deformation gradient, viz., $\mathbf{F}_{,1}$, located at `gaussPt` in configuration `state`.

`dFdY = p.dFdY(gaussPt, state)`

Returns the partial derivative taken with respect to the Y direction of a 2×2 planar deformation gradient, viz., $\mathbf{F}_{,2}$, located at `gaussPt` in configuration `state`.

`gMtx = p.G(gaussPt, state)`

Returns the 2×2 planar displacement gradient \mathbf{G} located at `gaussPt` in configuration `state`.

`qMtx = p.Q(gaussPt, state)`

Returns the 2×2 re-indexing matrix \mathbf{Q} located at `gaussPt` in configuration `state`.

`rMtx = p.R(gaussPt, state)`

Returns the 2×2 rotation matrix \mathbf{R} derived from a \mathbf{QR} decomposition of the re-indexed deformation gradient, denoted as \mathbf{RU} , located at `gaussPt` in configuration `state`.

`uMtx = p.U(gaussPt, state)`

Returns the 2×2 Laplace stretch \mathbf{U} derived from a \mathbf{QR} decomposition of the re-indexed deformation gradient located at `gaussPt` in configuration `state`.

`uInvMtx = p.UInv(gaussPt, state)`

Returns the inverse of a 2×2 Laplace stretch \mathbf{U}^{-1} derived from a \mathbf{QR} decomposition of the re-indexed deformation gradient located at `gaussPt` in configuration `state`.

`duMtx = p.dU(gaussPt, state)`

Returns a differential change in the 2×2 Laplace stretch \mathbf{U} derived from a \mathbf{QR} decomposition of the re-indexed deformation gradient located at `gaussPt` in configuration `state`.

`duInvMtx = p.dUInv(gaussPt, state)`

Returns a differential change in the inverse of a 2×2 Laplace stretch \mathbf{U}^{-1} derived from a \mathbf{QR} decomposition of the re-indexed deformation gradient located at `gaussPt` in configuration `state`.

Scalar attributes that are extensive thermodynamic variables, and their rates.

`xi = p.dilation(gaussPt, state)`

Returns the planar dilation derived from a \mathbf{QR} decomposition of the re-indexed deformation gradient located at `gaussPt` in configuration `state`, i.e., $\xi = \ln \sqrt{ab/a_0b_0}$.

`epsilon = p.squeeze(gaussPt, state)`

Returns the planar squeeze derived from a \mathbf{QR} decomposition of the re-indexed deformation gradient located at `gaussPt` in configuration `state`, i.e., $\varepsilon = \ln \sqrt{ab_0/a_0b}$.

`gamma = p.shear(gaussPt, state)`

Returns the planar shear derived from a \mathbf{QR} decomposition of the re-indexed deformation gradient located at `gaussPt` in configuration `state`.

`dDelta = p.dDilation(gaussPt, state)`

Returns differential change in the dilation of an irregular pentagon located at `gaussPt` in configuration `state`.

```
dEpsilon = p.dSqueeze(gaussPt, state)
```

Returns differential change in the squeeze of an irregular pentagon located at `gaussPt` in configuration `state`.

```
dGamma = p.dGamma(gaussPt, state)
```

Returns differential change in the shear of an irregular pentagon located at `gaussPt` in configuration `state`.

Appendix G. Constitutive Models

There are elastic constitutive models for 1D fibers, chordal fibers, 2D membranes, and 3D volumes. These are provided for in module `constitutiveEqns.py`.

Appendix G.1. Elastic Fibers

Five constitutive models are considered for 1D elastic fibers. Their mathematical representations are summarized in Eqn. (??). All five inherit the base class `elasticFibers` whose interface is:

class `elasticFiber`

implemented methods

These methods are intended to be called via a super call from all classes that extend class `elasticFiber`.

`<object>.__init__()`

This is the constructor, but it is not to be called externally, only internally from those classes that extend this base class.

`name = <object>.fiberType()`

Returns a string that contains the name of the fiber model.

`E = <object>.modulus(stress, strain, temperature)`

Returns the elastic tangent modulus at the specified `stress`, `strain` and `temperature`. It is the inverse of its elastic compliance.

virtual method

This method must be overridden by every fiber model that extends this base type. Virtual methods only provide an interface; their implementation is empty.

`C = <object>.compliance(stress, strain, temperature)`

Returns the elastic tangent compliance at the specified `stress`, `strain` and `temperature`.

Appendix G.1.1. Hookean Fibers

This class provides the elastic compliance and modulus for a Hookean fiber per Eqn. (??).

class `hooke(elasticFiber)`

constructor

`elasFiber = hooke(E)`

`E` the elastic modulus of the fiber, i.e., Young's modulus

```
name = elasFiber.fiberType()
```

Returns a string that contains the name of the fiber model, viz., 'Hooke'.

```
C = elasFiber.compliance(stress, strain, temperature)
```

Returns the elastic compliance at a specified **stress**, **strain** and **temperature**. None of these arguments are required.

```
E = elasFiber.modulus(stress, strain, temperature)
```

Returns the elastic modulus at a specified **stress**, **strain** and **temperature**, which is the inverse of its elastic compliance. None of these arguments are required.

Appendix G.1.2. Fungean Fibers

This class provides the elastic compliance and modulus for a Fungean fiber per Eqn. (??).

```
class fung1(elasticFiber)
```

constructor

```
elasFiber = fung1(E, beta)
```

E the elastic tangent modulus at zero stress and zero strain

beta strength of the exponential response

```
name = elasFiber.fiberType()
```

Returns a string that contains the name of the fiber model, viz., 'Fung'.

```
C = elasFiber.compliance(stress, strain, temperature)
```

Returns the elastic tangent compliance at a specified **stress**, **strain** and **temperature**. Only argument **stress** is required.

```
E = elasFiber.modulus(stress, strain, temperature)
```

Returns the elastic tangent modulus at a specified **stress**, **strain** and **temperature**, which is the inverse of its elastic compliance. Only argument **stress** is required.

Appendix G.1.3. Fungean/Hookean Fibers

This class provides the elastic response functions for Fungean and Hookean fibers whose compliances sum, per Eqn. (??).

```
class fung2(elasticFiber)
```

constructor

```
elasFiber = fung2(E1, E2, beta)
```

E1 the elastic tangent modulus at zero stress and zero strain

E2 the elastic tangent modulus at terminal stress

beta strength of the exponential response

```
name = elasFiber.fiberType()
```

Returns a string that contains the name of the fiber model, viz., 'Fung/Hooke'.

```
C = elasFiber.compliance(stress, strain, temperature)
```

Returns the elastic tangent compliance at a specified **stress**, **strain** and **temperature**. Only argument **stress** is required.

```
E = elasFiber.modulus(stress, strain, temperature)
```

Returns the elastic tangent modulus at a specified **stress**, **strain** and **temperature**, which is the inverse of its elastic compliance. Only argument **stress** is required.

Appendix G.1.4. Freed-Rajagopal Fibers

This class provides the elastic compliance and modulus for a Freed-Rajagopalean fiber per Eqn. (??).

```
class freed1(elasticFiber)
```

constructor

```
elasFiber = freed1(E, e_t)
```

E the elastic tangent modulus at zero stress and zero strain

e_t the limit strain, i.e., the maximum strain allowed

```
name = elasFiber.fiberType()
```

Returns a string that contains the name of the fiber model, viz., 'Freed-Rajagopal'.

```
C = elasFiber.compliance(stress, strain, temperature)
```

Returns the elastic tangent compliance at a specified **stress**, **strain** and **temperature**. Argument **temperature** is not required, but **stress** and **strain** are.

```
E = elasFiber.modulus(stress, strain, temperature)
```

Returns the elastic tangent modulus at a specified **stress**, **strain** and **temperature**, which is the inverse of its elastic compliance. Argument **temperature** is not required, but **stress** and **strain** are.

Appendix G.1.5. Freed-Rajagopal/Hookean Fibers

This class provides the elastic response functions for Freed-Rajagopalean and Hookean fibers whose compliances sum, per Eqn. (??). This version does not account for thermal straining.

```
class freed2(elasticFiber)
```

constructor

```
elasFiber = freed2(E1, E2, e_t)
```

E1 the elastic tangent modulus at zero stress and strain

E2 the elastic tangent modulus at terminal stress
e_t the limit strain, i.e., the maximum strain allowed

```
name = elasFiber.fiberType()
```

Returns a string that contains the name of the fiber model, viz., 'Freed-Rajagopal/Hooke'.

```
C = elasFiber.compliance(stress, strain, temperature)
```

Returns the elastic tangent compliance at a specified **stress** and **strain**. Argument **temperature** is not required, but **stress** and **strain** are.

```
E = elasFiber.modulus(stress, strain, temperature)
```

Returns the elastic tangent modulus at a specified **stress** and **strain**; it is the inverse of its elastic compliance. Argument **temperature** is not required, but **stress** and **strain** are.

Appendix G.1.6. Freed-Rajagopal/Kelvin/Hookean Fibers

This class provides the elastic response functions for Freed-Rajagopalean and Kelvin/Hookean fibers whose compliances sum, per Eqn. (??). This version does account for thermal straining.

```
class freed3(elasticFiber)
```

constructor

```
elasFiber = freed3(E1, E2, e_t, alpha, T0)
```

E1 the elastic tangent modulus at zero stress and strain
E2 the elastic tangent modulus at terminal stress
e_t the limit strain, i.e., the maximum strain allowed
alpha thermal strain coefficient
T0 the reference temperature for thermal strain, typically body temperature

```
name = elasFiber.fiberType()
```

Returns a string that contains the name of the fiber model, viz., 'Freed-Rajagopal/Kelvin/Hooke'.

```
C = elasFiber.compliance(stress, strain, temperature)
```

Returns the elastic tangent compliance at a specified **stress**, **strain** and **temperature**. All arguments are required.

```
E = elasFiber.modulus(stress, strain, temperature)
```

Returns the elastic tangent modulus at a specified **stress**, **strain** and **temperature**, which is the inverse of its elastic compliance. All arguments are required.

Appendix G.2. Chordal Fibers

Appendix H. Solvers

Two ODE solvers are included in this software. The first, `peceVtoX.py`, uses a two-step PECE (Predict, Evaluate, Correct, re-Evaluate) method to solve a first-order, ordinary differential equation $\dot{\mathbf{x}}(t) = \mathbf{v}(t) = \mathbf{f}(t, \mathbf{x})$ given an initial condition $\mathbf{x}_0 = \mathbf{x}(t_0)$ where, as an analogy, \mathbf{x} denotes displacement and \mathbf{v} denotes velocity. The second, `peceAtoVandX.py`, uses another two-step PECE method to solve a second-order, ordinary, differential equation $\ddot{\mathbf{x}}(t) = \mathbf{a}(t) = \mathbf{f}(t, \mathbf{x}, \mathbf{v})$ given initial conditions $\mathbf{x}_0 = \mathbf{x}(t_0)$ and $\mathbf{v}_0 = \mathbf{v}(t_0) = \dot{\mathbf{x}}(t_0)$ where, as an analogy, \mathbf{x} denotes displacement, \mathbf{v} denotes velocity, and \mathbf{a} denotes acceleration.

Appendix H.1. 1st Order ODE Solver

Module `peceVtoX.py` is a Python code that exports class `pece` which solves first-order, ordinary, differential equations using a two-step method; in particular, it solves

$$\mathbf{v} = \mathbf{f}(t, \mathbf{x}) \quad \text{where} \quad \mathbf{v} = \dot{\mathbf{x}} \quad \text{satisfying IC} \quad \mathbf{x}_0 = \mathbf{x}(t_0)$$

where the dependent variables of integration \mathbf{x} are analogous to displacements, while the ODEs $\dot{\mathbf{x}} = \mathbf{f}(t, \mathbf{x})$ are analogous to velocities $\mathbf{v} = \dot{\mathbf{x}}$.

class `pece`

constructor

```
solver = pece(ode, t0, x0, h, tol=0.0001)
    ode  the differential equation to be solved, i.e.,  $\dot{\mathbf{x}} = \mathbf{v} = \mathbf{f}(t, \mathbf{x})$  where  $\text{ode} = \mathbf{f}(t, \mathbf{x})$ 
    t0   the initial time  $t$ , viz., time at the start of integration
    x0   the initial condition, viz., displacements at the start of integration  $\mathbf{x}_0 = \mathbf{x}(t_0)$ 
    h    the global time-step size separating two neighboring states
    tol  the maximum allowed local truncation error, with a default set at  $10^{-4}$ 
```

methods

```
solver.integrate()
```

A command that integrates the ODE from current time t_n to the next time $t_{n+1} = t_n + h$. This command may be called multiple times before committing a solution. A local time stepper is used to integrate over the global time step. The local time-step size is controlled by a PI controller that runs in the background. This controller bounds the local truncation error from above. If the error is too small then the controller increases the local step size. If the error is too large then the controller decreases the local step size.

```
solver.advance()
```

A command that updates the internal data structure of the integrator by relabeling variables assigned to current time t_n to their counterparts associated with previous time t_{n-1} , and then assigning the variables just solved for at time t_{n+1} to their counterparts at time t_n . This performs an incremental advancement of the solution along its trajectory, with $t_n + h$ now becoming the current time.

The following methods are to be called after a solution has been advanced/committed, but before the next integration step is taken.

```
n, nd, nh, nr = solver.getStatistics()
    n    total number of local steps taken
    nd    total number of local steps taken where the step-size was doubled
    nh    total number of local steps taken where the step-size was halved
    nr    total number of local steps taken where the integrator was restarted
```

```
t = solver.getT()
```

Returns the current time **t**, i.e., the independent variable of integration.

```
x = solver.getX()
```

Returns the solution vector **x** at current time, i.e., the dependent variables of integration.

```
v = solver.getV()
```

Returns the time rate-of-change in the dependent variables at current time, i.e., the ODEs being solved, their analog being velocities.

```
err = solver.getError()
```

Returns an estimate for the local truncation error **err** at current time.

```
x = solver.interpolate(atT)
```

Returns the solution **x(atT)** at time **atT** using cubic Hermite interpolation, where **atT** is located somewhere between the previous t_{n-1} and current t_n times of the integrator.

Appendix H.2. 2nd Order ODE Solver

Module **peceAtoVandX.py** is a Python code that exports class **pece** which solves second-order, ordinary, differential equations using a two-step method; in particular, it solves

$$\mathbf{a} = \mathbf{f}(t, \mathbf{x}, \mathbf{v}) \quad \text{where} \quad \mathbf{a} = \ddot{\mathbf{x}} \quad \text{and} \quad \mathbf{v} = \dot{\mathbf{x}} \quad \text{with ICs} \quad \mathbf{x}_0 = \mathbf{x}(t_0) \quad \text{and} \quad \mathbf{v}_0 = \mathbf{v}(t_0)$$

where the dependent variables of integration **x** are analogous to displacements whose rates $\mathbf{v} = \dot{\mathbf{x}}$ are analogous to velocities, while the ODEs $\ddot{\mathbf{x}} = \mathbf{f}(t, \mathbf{x}, \mathbf{v})$ are analogous to accelerations $\mathbf{a} = \dot{\mathbf{v}} = \ddot{\mathbf{x}}$.

This solver is useful when solving dynamics problems, e.g.,

$$\mathbf{M}\mathbf{a} + \mathbf{C}\mathbf{v} + \mathbf{K}\mathbf{x} = \boldsymbol{\phi}(t, \mathbf{x}, \mathbf{v}) \quad \text{or} \quad \mathbf{a} = \mathbf{f}(t, \mathbf{x}, \mathbf{v}) \quad \text{with} \quad \mathbf{f} = \mathbf{M}^{-1}(\boldsymbol{\phi}(t) - \mathbf{C}\mathbf{v} - \mathbf{K}\mathbf{x})$$

where **M** is a mass matrix, **C** is a damping matrix, **K** is a stiffness matrix, and $\boldsymbol{\phi}$ is a forcing function. Typically **M** is diagonal so its inverse is trivial.

class pece

constructor

solver = pece(aFn, t0, x0, v0, h, tol=0.0001)

aFn the differential equation to be solved, i.e., $\ddot{\mathbf{x}} = \mathbf{a} = \mathbf{f}(t, \mathbf{x}, \dot{\mathbf{x}})$ where $\mathbf{aFn} = \mathbf{f}(t, \mathbf{x}, \mathbf{v})$

t0 the initial time t , viz., time at the start of integration

x0 an initial condition, viz., displacements at the start of integration $\mathbf{x}_0 = \mathbf{x}(t_0)$

v0 an initial condition, viz., velocities at the start of integration $\mathbf{v}_0 = \mathbf{v}(t_0) = \dot{\mathbf{x}}(t_0)$

h the global time-step size separating two neighboring states

tol the maximum allowed local truncation error, with a default set at 10^{-4}

methods

solver.integrate()

A command that integrates the ODE from current time t_n to the next time $t_{n+1} = t_n + h$. This command may be called multiple times before committing a solution. A local time stepper is used to integrate over the global time step. The local time-step size is controlled by a PI controller that runs in the background. This controller bounds the local truncation error from above. If the error is too small then the controller increases the local step size. If the error is too large then the controller decreases the local step size.

solver.advance()

A command that updates the internal data structure of the integrator by relabeling variables assigned to current time t_n to their counterparts associated with previous time t_{n-1} , and then assigning the variables just solved for at time t_{n+1} to their counterparts at time t_n . This performs an incremental advancement of the solution along its trajectory, with $t_n + h$ now becoming the current time.

The following methods are to be called after a solution has been advanced/committed, but before the next integration step is taken.

n, nd, nh, nr = solver.getStatistics()

n total number of local steps taken

nd total number of local steps taken where the step-size was doubled

nh total number of local steps taken where the step-size was halved

nr total number of local steps taken where the integrator was restarted

t = solver.getT()

Returns the current time **t**, i.e., the independent variable of integration.

x = solver.getX()

Returns the solution vector **x** at current time, i.e., first set of the dependent variables of integration.

`v = solver.getV()`

Returns the solution vector `v` at current time, i.e., second set of the dependent variables of integration.

`a = solver.getA()`

Returns the time rate-of-change in the velocity variables at current time, i.e., the ODEs being solved, their analog being accelerations.

`err = solver.getError()`

Returns an estimate for the local truncation error `err` at current time.

`x = solver.interpolateX(atT)`

Returns the solution `x(atT)` at time `atT` using cubic Hermite interpolation, where `atT` is located somewhere between the previous t_{n-1} and current t_n times of the integrator.

`x = solver.interpolateV(atT)`

Returns the solution `v(atT)` at time `atT` using cubic Hermite interpolation, where `atT` is located somewhere between the previous t_{n-1} and current t_n times of the integrator.