Brain Tumor Detection: Linear Elastic Model

Anil Aksu,
anil.aksu@tubitak.gov.tr

Ozgur Gundogan
ozgurgundogan90@gmail.com

September 8, 2017

Contents

Chapter 1

Introduction

In the present work, 2D model of the possible anomaly detection in elastic media is analysed. The main idea is to excite the elastic media from two distinct location and measure the stress waves at the similar locations. The main objective is to estimate the location of the possible anomaly in the media based on the differences of stess distributions on sensors. It is done both analytically and numerically. After generating enough of data set, it is planned to take advantage of machine learning algorithms.

1.1 Problem Description

In a simple 2D elastodynamics problem, the media without any anomaly can be given as a rectangle for baseline analysis. The purpose is to obtain first order analytical model then extend it to a problem with an anomaly at boundaries. In the analysis to be described in next chapters, these anomalies assumed to have small characteristic length h compared to characteristic length of the whole domain which is the domain L where $h/L = \epsilon << 1$. This will allow us to perform perturbation analysis. So far, it is called anomaly[2], however in real life, these anomalies are brain tumors locate on the inner surface of the skull[3]. This model is just representative of it.

Moreover, in a practical application, this elastic media is excited on two different locations and the stress distribution is also measured in two different locations as well. The purpose of using two different excitation regions and two different receiver points is to mimic the parallax effect in human eyes which can perceive three dimensional space with two distinct two dimensional view. The perception of depth in human vision is dependent on the differences between these two distinct views. In our case, these two distinct views are replaced with two distinct stress or strain distribution. The ultimate objective is to reconstruct the field based on these two distinct stress distribution which is explained schematically in figure 1.

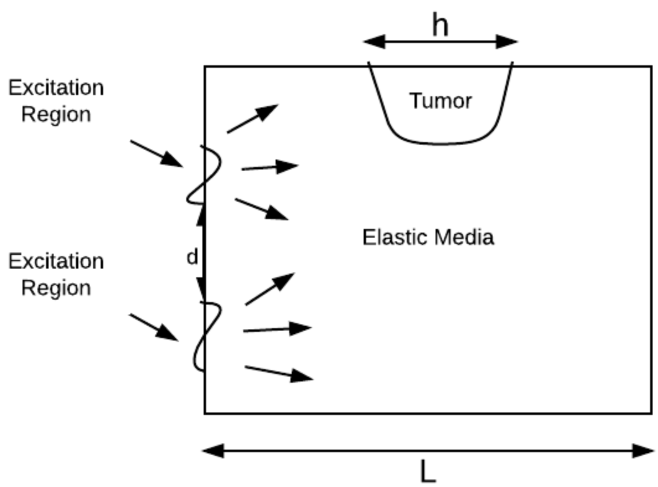

Figure 1.1: Schematic Illustration of the anomaly detection problem where a tumor is located on the top boundary.

The equation of motion in the elastic media[**?**] given in figure 1 can be simply obtained by Newton's law as:

$$\rho \frac{\partial^2 u_i}{\partial t^2} = \frac{\partial \tau_{ij}}{\partial x_j}. \tag{1.1}$$

where ρ is the density, u_i is the deformation vector, τ_{ij} is the stress tensor. After replacing the stress tensor with the definition given in equation A.8. The resultant equations of motion can be given as:

$$\rho \frac{\partial^2 u_x}{\partial t^2} = ((2\mu + \lambda)\frac{\partial^2}{\partial x^2} + \mu \frac{\partial^2}{\partial y^2})u_x + ((\mu + \lambda)\frac{\partial^2}{\partial x \partial y})u_y, \tag{1.2}$$

$$\rho \frac{\partial^2 u_y}{\partial t^2} = ((\mu + \lambda)\frac{\partial^2}{\partial x \partial y})u_x + ((2\mu + \lambda)\frac{\partial^2}{\partial y^2} + \mu \frac{\partial^2}{\partial x^2})u_y, \tag{1.3}$$

where λ and μ are Lame constants. These equations of motion given above are derived in 2-D, it can also be extended to 3-D problem. However, before attacking more complicated problem, its validity and applications will be tested. Moreover, the equations above can sustain the wave behaviour, therefore there is a fundamental frequency and associated wave number vectors in x and y directions. These can be related by so-called **dispersion relation**. The dispersion relation can be derived by assuming $u_x = A_x e^{i(k_x x + k_y y - \omega t)}$ and $u_y = A_y e^{i(k_x x + k_y y - \omega t)}$. After replacing them into equation 1.2 and equation 1.3, the following linear system is obtained:

$$\begin{bmatrix} -(2\mu+\lambda)k_x^2 - \mu k_y^2 + \rho\omega^2 & -(\mu+\lambda)k_x k_y \\ -(\mu+\lambda)k_x k_y & -(2\mu+\lambda)k_y^2 - \mu k_x^2 + \rho\omega^2 \end{bmatrix} \begin{bmatrix} A_x \\ A_y \end{bmatrix} = 0. \quad (1.4)$$

To have non-trivial solution to u_x and u_y, the determinant of the matrix in equation 1.4 must vanish, therefore the dispersion relation is obtained as:

$$\rho^2\omega^4 - (3\mu+\lambda)(k_x^2+k_z^2)\rho\omega^2 + (2\mu^2+\lambda\mu)(k_x^4+k_z^4) + (4\mu^2+2\lambda\mu)k_x^2k_z^2 = 0. \quad (1.5)$$

The roots of the dispersion relation given in equation 1.5 are given in Appendix B. This relation play an important role on determining the analytical solution and the angle of propagation from the excitation region.

Chapter 2

Analytical Solution

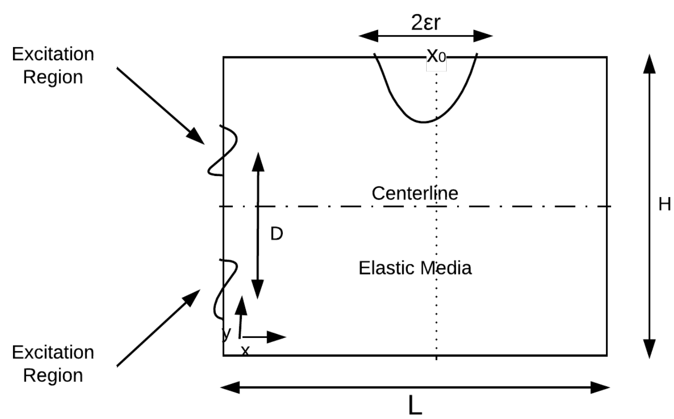

Figure 2.1: The solution domain where a relatively small semi-circular anomaly is located at the top boundary with a raduis ϵr at x_0.

The solution can be represented in terms of irrotational vector and solenoidal part as:

$$\mathbf{u} = \nabla\phi + \nabla \times \mathbf{H}. \tag{2.1}$$

where

$$\nabla \cdot \mathbf{H} = 0. \tag{2.2}$$

After replacing these into equations 1.2 and 1.3, the following set of separate equations are obtained:

$$\rho\frac{\partial^2\phi}{\partial t^2} = (\lambda + 2\mu)\nabla^2\phi, \tag{2.3}$$

and

$$\rho\frac{\partial^2\mathbf{H}}{\partial t^2} = \mu\nabla^2\mathbf{H}. \tag{2.4}$$

The equations 2.1, 2.2, 2.3 and 2.4 are basically wave equation in 2-D. In general, the wave equation is given as:

$$u_{tt} = c^2(u_{xx} + u_{yy}). \tag{2.5}$$

It is solved easily by separation of variables by introducing $u(x, y, t) = X(x)Y(y)T(t)$. If it is replaced into equation 2.5, the following equation is obtained:

$$XYT'' = c^2(X''YT + XY''T). \tag{2.6}$$

Let's divide equation 2.6 by XYT, the following equation is obtained:

$$\frac{T''}{T} = c^2(\frac{X''}{X} + \frac{Y''}{Y}). \tag{2.7}$$

In eigenfunction form, the equation 2.7 can be given as a set of three equations:

$$T'' + \lambda_{nt}^2 T = 0, \tag{2.8}$$

$$X'' + \lambda_{nx}^2 X = 0, \tag{2.9}$$

and

$$Y'' + \lambda_{ny}^2 Y = 0. \tag{2.10}$$

where

$$\lambda_{nt}^2 = c^2(\lambda_{nx}^2 + \lambda_{ny}^2). \tag{2.11}$$

Therefore, the solution can be given as:

$$\phi(x, y, t) = \sum_{nx=1}^{N_x}\sum_{ny=1}^{N_y}(\alpha_{nx,ny}^s \sin\lambda_{nt}t + \alpha_{nx,ny}^c \cos\lambda_{nt}t)\sin\lambda_{nx}x \sin\lambda_{ny}y. \tag{2.12}$$

therefore

$$u_x(x, y, t) = \sum_{nx=1}^{N_x}\sum_{ny=1}^{N_y}\lambda_{nx}(\alpha_{nx,ny}^s \sin\lambda_{nt}t + \alpha_{nx,ny}^c \cos\lambda_{nt}t)\cos\lambda_{nx}x \sin\lambda_{ny}y. \tag{2.13}$$

and

$$u_y(x, y, t) = \sum_{nx=1}^{N_x}\sum_{ny=1}^{N_y}\lambda_{ny}(\alpha_{nx,ny}^s \sin\lambda_{nt}t + \alpha_{nx,ny}^c \cos\lambda_{nt}t)\sin\lambda_{nx}x \cos\lambda_{ny}y. \tag{2.14}$$

In this configuration, it is assumed that there is no displacement on boundaries, therefore the displacement field can be given as:

$$\vec{u} \cdot \vec{n} = 0. \tag{2.15}$$

where $\vec{u}_0 + \epsilon\vec{u}_1$ and also the unit vector $\vec{n}$ is defined as:

$$\vec{n} = \nabla\Omega. \tag{2.16}$$

where $\Omega(x, y) = y - \varepsilon e^{-\frac{(x-x_0)^2}{2\sigma^2}}$. Note that $\varepsilon << 1$ therefore it is assumed that the anomaly height is small compared to the domain height H. As a result, the unit vector can be given as:

$$\vec{n} = \hat{y} + \varepsilon\frac{(x - x_0)}{\sigma^2}e^{-\frac{(x-x_0)^2}{2\sigma^2}}\hat{x}. \tag{2.17}$$

Therefore the condition 2.15 results in the following conditions as powers of ε. At the leading order $O(1)$:

$$u_y^0 = 0, \tag{2.18}$$

At the order $O(\epsilon)$:

$$u_y^1 + u_x^0\frac{(x - x_0)}{\sigma^2}e^{-\frac{(x-x_0)^2}{2\sigma^2}} = 0. \tag{2.19}$$

At the leading order, the boundary value problem can be described as $u_y^0(x, H/2, t) = u_y^0(x, -H/2, t) = 0$, $u_x^0(L/2, y, t) = 0$ and $u_x^0(-L/2, y, t) = f(y, t)$ where $f(y, t)$ is the excitation force which can be described as:

$$f(y,t) = \begin{cases} 0 & D/2 + \lambda/2 < y \leq L/2 \\ A\sin(k_y y - \omega t) & D/2 - \lambda/2 > y > D/2 + \lambda/2 \\ 0 & D/2 - \lambda/2 > y > -D/2 + \lambda/2 \\ A\sin(k_y y - \omega t) & -D/2 + \lambda/2 > y > -D/2 - \lambda/2 \\ 0 & -D/2 - \lambda/2 > y \geq L/2 \end{cases} \tag{2.20}$$

Table 2.1: Due to compatibility of initial conditions with boundary conditions, $\partial_t u_x^0(x,y,0)$ and $\partial_t u_y^1(x,y,0)$ are not set to zero.

Initial Conditions			
ICs at $O(1)$	$u_x^0(x,y,0)=0$	$u_y^0(x,y,0)=0$	$\partial_t u_y^0(x,y,0)=0$
ICs at $O(\varepsilon)$	$u_x^1(x,y,0)=0$	$u_y^1(x,y,0)=0$	$\partial_t u_x^1(x,y,0)=0$

Table 2.2: Due to compatibility of initial conditions with boundary condti

Boundary Conditions		
Order of BC	$O(1)$	$O(\varepsilon)$
BC at $x=-L/2$	$u_x^0(-L/2,y,t)=f(y,t)$	$u_x^1(-L/2,y,t)=0$
BC at $x=L/2$	$u_x^0(L/2,y,t)=0$	$u_x^1(L/2,y,t)=0$
BC at $y=-H/2$	$u_y^0(x,-H/2,t)=0$	$u_y^1(x,-H/2,t)=0$
BC at $y=H/2$	$u_y^0(x,H/2,t)=0$	$u_y^1(x,H/2,t)=-u_x^0\frac{\mathrm{d}h}{\mathrm{d}x}$

Under these boundary conditions, the eigenvalues in y direction can be found as:

$$\lambda_{ny}=\frac{\pi}{H}+\frac{2(n-1)\pi}{H}. \tag{2.21}$$

Note that the eigenvalue in x direction has to satisfy the dispersion relation, therefore it can be given as:

$$\lambda_{nx}=\sqrt{\frac{\omega^2}{\lambda}-\lambda_{ny}^2} \tag{2.22}$$

The right boundary condition at the leading order requires that

$$\cos(\lambda_{nx}L+\phi_0)=0. \tag{2.23}$$

where $\phi_0^{nx}=\pi/2-\lambda_{nx}L$ is the phase shift at each mode. The excitation is imposed on left boundary on u_x which can be decomposed in t

Chapter 3

Numerical Solution

Even though the physical model is obtained, solving the governing system analytically is not practical in most cases. Therefore an appropriate numerical model is required to get results for arbitrary geometry of anomaly[1]. Before discussing the geometry of the problem, let's first introduce the numerical model in the problem. Spectral methods are preferred in this problem, which gives exponential convergence rate. It has the same principle with the finite element method which has basis functions so-called shape functions, in the case of spectral methods, the basis functions are in general Fourier series or Legendre polynomials. But in the specific problem analysed here, Legendre polynomials of first kind is preferred. As a result, the displacement field can be given as:

$$u_x = \sum_{i=1}^{N_x} \sum_{j=1}^{N_y} \alpha_{ij}^x(t) P_i(x) P_j(y), \tag{3.1}$$

$$u_y = \sum_{i=1}^{N_x} \sum_{j=1}^{N_y} \alpha_{ij}^y(t) P_i(x) P_j(y). \tag{3.2}$$

The main advantage of the problem stems from the functional orthogonality of the basis functions. Both Fourier series and Legendre polynomial are orthogonal set of functions on their mother interval $x \in [-1, 1]$, therefore:

$$< P_n(x) P_m(x) > \int_{-1}^{1} P_n(x) P_m(x) \mathrm{d}x = \frac{2}{2+n} \delta_{mn}. \tag{3.3}$$

The orthogonality property is commonly used to reduce the error and the computational effort by means of weak formulation. In case of weak formulation, equation 1.1 and equation 1.2 are multiplied with $P_i(x)P_j(y)$ and integrated over whole domain. As a result, the mass matrix $[\mathbf{M}]$ and the stiffness matrix $[\mathbf{K}]$, explicitly given in Appendix C, are obtained for displacements in both x and y directions separately as:

$$[\mathbf{M}]\alpha_{tt} = [\mathbf{K}]\alpha. \tag{3.4}$$

The mass matrix and the stiffness matrix should be defined explicitly. To start with, the mass matrix is formed by weak formulation of the problem given above, it appears on the left hand side of the equation which governs the time derivative. It can be explicitly given as:

$$[\mathbf{M}] = \begin{bmatrix} [\mathbf{M}^x] & 0 \\ 0 & [\mathbf{M}^y] \end{bmatrix} \tag{3.5}$$

Note that $[\mathbf{M}^x]$ and $[\mathbf{M}^y]$ are mass matrices for u_x and u_y respectively. Since they are evaluated in the same domain, they are equal to each other, therefore $[\mathbf{M}^x] = [\mathbf{M}^y]$. Explicitly, the mass matrix for u_x can be given as:

$$M^x_{((j-1)\times N_x+i),((j-1)\times N_x+i)} = \rho w^x_i w^y_j \frac{L}{2}\frac{H}{2}. \tag{3.6}$$

Note that N_x and N_y are the number of points in x and y directions and w^x_i and w^y_j are quadrature weights on Legendre Gauss Lobatto points. Furthermore, L is the domain length and H is the domain height, they appear due to coordinate mapping from $[-1, 1]$ to $[0, L]$ and $[0, H]$. In a similar fashion, the stiffness matrix can also be written. However, the differentiation matrices should be written first. Let's denote $[\mathbf{D}_x]$ as the differentiation matrix in x direction and $[\mathbf{D}_y]$ as the differentiation matrix in y direction. Under this formulation, the stiffness matrix can be given as:

$$[\mathbf{K}] = \begin{bmatrix} -(\mu+\frac{\lambda}{2})[\mathbf{D}^x]^T[\mathbf{D}^x] - \frac{\mu}{2}[\mathbf{D}^y]^T[\mathbf{D}^y] & (\mu+\lambda)[\mathbf{D}^x][\mathbf{D}^y] \\ (\mu+\lambda)[\mathbf{D}^x][\mathbf{D}^y] & -(\mu+\frac{\lambda}{2})[\mathbf{D}^x]^T[\mathbf{D}^x] - \frac{\mu}{2}[\mathbf{D}^y]^T[\mathbf{D}^y] \end{bmatrix} \tag{3.7}$$

After giving the mass matrix and the stiffness matrix explicitly, they can be replaced in equation 3.4. However, the second time derivative still appears explicitly in equation 3.4, it can be reduced to the first order time derivative if the system is converted to state-space form by introducing a new set of variables as:

$$\alpha_t = [\mathbf{I}]\beta. \tag{3.8}$$

After combining equation 3.4 and equation 3.8 and inverting the matrix formed by combining mass matrix and identity matrix in equation 3.8, it can be transformed into the following system of equations as:

$$\begin{bmatrix} \beta_t \\ \alpha_t \end{bmatrix} = \begin{bmatrix} [\mathbf{M}] & 0 \\ 0 & [\mathbf{I}] \end{bmatrix}^{-1} \begin{bmatrix} 0 & [\mathbf{K}] \\ [\mathbf{I}] & 0 \end{bmatrix} \begin{bmatrix} \beta \\ \alpha \end{bmatrix}. \tag{3.9}$$

So far the spatial discretization and the matrix notation are introduced, however the system should be integrated in time to obtain transient solution. Before proceeding to time integration, let's call the system matrix [**G**] which can be explicitly given as:

$$[\mathbf{G}] = \begin{bmatrix} [\mathbf{M}] & 0 \\ 0 & [\mathbf{I}] \end{bmatrix}^{-1} \begin{bmatrix} 0 & [\mathbf{K}] \\ [\mathbf{I}] & 0 \end{bmatrix}. \tag{3.10}$$

For accuracy and stability, semi-implicit time integration method is preferred which is Crank-Nicholson. This time integration scheme can be given as:

$$\begin{bmatrix} \beta \\ \alpha \end{bmatrix}^{n+1} - \begin{bmatrix} \beta \\ \alpha \end{bmatrix}^{n} = \Delta t\theta[\mathbf{G}] \begin{bmatrix} \beta \\ \alpha \end{bmatrix}^{n} + \Delta t(1-\theta)[\mathbf{G}] \begin{bmatrix} \beta \\ \alpha \end{bmatrix}^{n+1}. \tag{3.11}$$

Chapter 4

Machine Learning Algorithm: 2-D surface construction from 1-D data

Algorithm 1 Semi Implicit Time Integration

procedure SEMI IMPLICIT TIME INTEGRATION

    $T(i) \leftarrow$ Temperature of each node at previous time step
    $N_{time} \leftarrow$ number of time steps
    $K_{ij} \leftarrow$ Get the thermal conductance between each node
    $Q_{sun} \leftarrow$ Get heat flux from Sun onto each surface
    $R_{ij} \leftarrow$ Get the radiation conductor between each node
    $\mathrm{d}t \leftarrow$ time step size

    while $L_\infty(error) > 10^{-2}$ **do**
        $Coordinate \leftarrow$ Satellite location and orientation

        $\vec{n}_s \leftarrow$ Calculate unit normal of Sun based on satellite location
        $Q_{sun} \leftarrow$ Calculate heat flux from Sun onto each surface
        $R_{ij} \leftarrow$ Calcualte the radiation conductor between each node

        $T(i+1) \leftarrow$ Semi-Implicit Time Integration using $T(i)$, R_{ij}, K_{ij} and $\mathrm{d}t$
    end while
end procedure

Chapter 5

Results

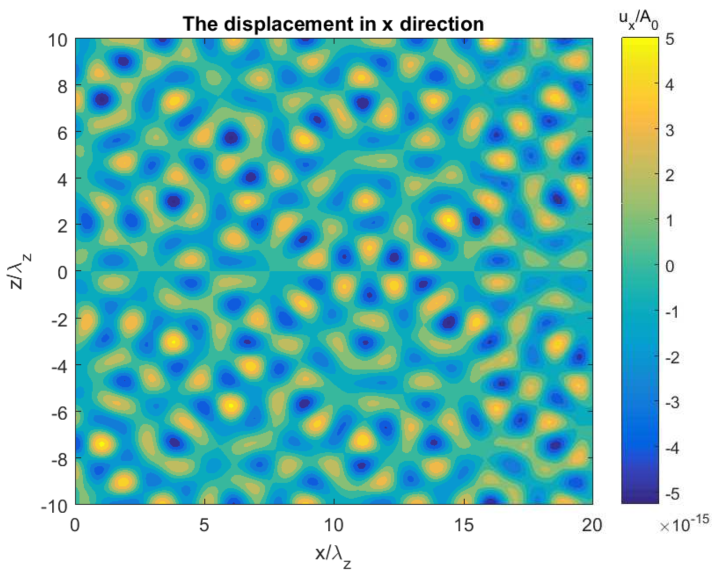

Figure 5.1: The displacement fields n^{th} order response under boundary forcing.

Chapter 6

Discussion

Appendix A

Basics of Elasticity

The coordinate system is defined in terms of both reference $\mathbf{X}$ and deformed states $\mathbf{x}$ and they are associated each other by the deformation vector $\mathbf{u}$ as:

$$\mathbf{x} = \mathbf{X} + \mathbf{u}. \tag{A.1}$$

where $\mathbf{u}$ is a function of the reference coordinate system. The infinitesimal deformation is given as:

$$[\mathrm{d}\mathbf{x}] = ([I] + [\frac{\partial \mathbf{u}}{\partial \mathbf{X}}])[\mathrm{d}\mathbf{X}]. \tag{A.2}$$

Under these conditions, the extension in terms of arclength is given as:

$$[\mathrm{d}\mathbf{x}]^T[\mathrm{d}\mathbf{x}] = [\mathrm{d}\mathbf{X}]^T([I] + [\frac{\partial \mathbf{u}}{\partial \mathbf{X}}])^T([I] + [\frac{\partial \mathbf{u}}{\partial \mathbf{X}}])[\mathrm{d}\mathbf{X}]. \tag{A.3}$$

In equation A.3, the relation between the initial arclength and the deformed arclength is given. They are related via Cauchy-Green strain tensor. In indicial notation, it is given as:

$$C_{ij} = \delta_{ij} + (\frac{\partial u_i}{\partial X_j} + \frac{\partial u_j}{\partial X_i}) + \frac{\partial u_m}{\partial X_i}\frac{\partial u_m}{\partial X_j}. \tag{A.4}$$

However, for small deformations, the non-linear term is negligible, therefore the equation A.4 can be reduced to the linear form as:

$$C_{ij} = \delta_{ij} + (\frac{\partial u_i}{\partial X_j} + \frac{\partial u_j}{\partial X_i}). \tag{A.5}$$

Therefore, the strain tensor can be given in terms of Cauchy-Green strain tensor as:

$$e_{ij} = \frac{1}{2}(C_{ij} - \delta_{ij}) = \frac{1}{2}(\frac{\partial u_i}{\partial X_j} + \frac{\partial u_j}{\partial X_i}). \tag{A.6}$$

The stress can also be defined in terms of stress as:

$$\tau_{ij} = \lambda e_{kk}\delta_{ij} + 2\mu e_{ij}. \tag{A.7}$$

In two dimension, it can be given explicitly as:

$$\begin{bmatrix} \tau_{xx} & \tau_{xy} \\ \tau_{yx} & \tau_{yy} \end{bmatrix} = \begin{bmatrix} 2\mu\frac{\partial u_1}{\partial X_1} + \lambda(\frac{\partial u_1}{\partial X_1} + \frac{\partial u_2}{\partial X_2}) & \mu(\frac{\partial u_1}{\partial X_2} + \frac{\partial u_2}{\partial X_1}) \\ \mu(\frac{\partial u_1}{\partial X_2} + \frac{\partial u_2}{\partial X_1}) & 2\mu\frac{\partial u_2}{\partial X_2} + \lambda(\frac{\partial u_1}{\partial X_1} + \frac{\partial u_2}{\partial X_2}) \end{bmatrix} \tag{A.8}$$

A.1 Root of Dispersion Relation

The dispersion relation is obtained as:

$$\rho^2\omega^4 - (3\mu+\lambda)(k_x^2+k_z^2)\rho\omega^2 + (2\mu^2+\lambda\mu)(k_x^4+k_z^4) + (4\mu^2+2\lambda\mu)k_x^2k_z^2 = 0. \tag{A.9}$$

More compactly, it can be given as:

$$\omega^4 + \alpha\omega^2 + \beta = 0. \tag{A.10}$$

where $\alpha = -(3\mu + \lambda)(k_x^2 + k_z^2)/\rho$ and $\beta = ((2\mu^2 + \lambda\mu)(k_x^4 + k_z^4) + (4\mu^2 + 2\lambda\mu)k_x^2k_z^2)/\rho^2$. As a first step, let's find roots for ω^2, then ω roots can be obtained much simpler. Therefore;

$$\omega_{1,2}^2 = -\frac{\alpha \pm \sqrt{\alpha^2 - 4\beta}}{2}. \tag{A.11}$$

Therefore, the four roots are obtained as:

$$\begin{aligned} \omega_{1,2,3,4} = & -\frac{\sqrt{-\alpha - \sqrt{\alpha^2 - 4\beta}}}{\sqrt{2}}, -\frac{\sqrt{-\alpha + \sqrt{\alpha^2 - 4\beta}}}{\sqrt{2}}, \\ & \frac{\sqrt{-\alpha - \sqrt{\alpha^2 - 4\beta}}}{\sqrt{2}}, \frac{\sqrt{-\alpha + \sqrt{\alpha^2 - 4\beta}}}{\sqrt{2}}. \end{aligned} \tag{A.12}$$

A.2 Fourier Series Form of Boundary Forcing

$$f(y,t) = \begin{cases} 0 & D/2 + \lambda/2 < y \leq L/2 \\ A\sin(k_y y - \omega t) & D/2 - \lambda/2 > y > D/2 + \lambda/2 \\ 0 & D/2 - \lambda/2 > y > -D/2 + \lambda/2 \\ A\sin(k_y y - \omega t) & -D/2 + \lambda/2 > y > -D/2 - \lambda/2 \\ 0 & -D/2 - \lambda/2 > y \geq L/2 \end{cases} \tag{A.13}$$

The boundary forcing A.13 can be decomposed as follows:

$$f_1(y)\sin\omega t + f_2(y)\cos\omega t. \tag{A.14}$$

where

$$f_1(y) = \begin{cases} 0 & D/2+\lambda/2 < y \le L/2 \\ -A\cos(k_y y) & D/2-\lambda/2 > y > D/2+\lambda/2 \\ 0 & D/2-\lambda/2 > y > -D/2+\lambda/2 \\ -A\cos(k_y y) & -D/2+\lambda/2 > y > -D/2-\lambda/2 \\ 0 & -D/2-\lambda/2 > y \ge L/2 \end{cases} \tag{A.15}$$

and

$$f_2(y) = \begin{cases} 0 & D/2+\lambda/2 < y \le L/2 \\ A\sin(k_y y) & D/2-\lambda/2 > y > D/2+\lambda/2 \\ 0 & D/2-\lambda/2 > y > -D/2+\lambda/2 \\ A\sin(k_y y) & -D/2+\lambda/2 > y > -D/2-\lambda/2 \\ 0 & -D/2-\lambda/2 > y \ge L/2 \end{cases} \tag{A.16}$$

The eigenvalues in y direction are found as:

$$\lambda_{ny} = \frac{\pi}{H} + \frac{2(n-1)\pi}{H}. \tag{A.17}$$

$$\alpha^s_{nx,ny} = \frac{2}{L}\int_{-H/2}^{H/2} f_1(y)\sin\lambda_{ny}y\mathrm{d}y = 0, \tag{A.18}$$

$$\begin{aligned} \alpha^c_{nx,ny} &= \frac{2}{L}\int_{-H/2}^{H/2} f_2(y)\sin\lambda_{ny}y\mathrm{d}y \\ &= \frac{-2A}{L}(\cos((\lambda_{ny}+k_y)(D/2-\lambda/2)) - \cos((\lambda_{ny}+k_y)(D/2+\lambda/2)) + \\ &\quad \cos((\lambda_{ny}-k_y)(D/2+\lambda/2)) - \cos((\lambda_{ny}-k_y)(D/2-\lambda/2)), \end{aligned} \tag{A.19}$$

A.3 2-D Analytical Elastodynamic Solver

```
% created by Anil Aksu
% this routine is generated to plot the analytical solution for ...
    the small
% disturbance on the top of rectangular domain which is meant to ...
    model a
% brain tumor

clear all
```

```
format long

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                                         %
%          Excitation Parameters          %
%                                         %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% the freqency of the excitation
omega=0.176;
% the wave length in z direction
lamz=0.875/2.;
% the wave number in z direction
kz=2*pi/lamz;
% the tumor amplitude
A=0.1;
% the distance between the excitation regions
D=8.*lamz;


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                                         %
%             Grid Parameters             %
%                                         %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% the number of points in x direction
Nx=200;
% the number of points in y direction
Ny=200;

% the length of the domain
L=20*lamz;
H=20*lamz;
% the location of tumor
x_0=L/2.;

% the mother interval
x_mother=linspace(0,L,Nx);
y_mother=linspace(-H/2,H/2,Ny);


%% the grid
x_grid=zeros(Nx*Ny,2);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                                         %
%        Wave Energetics Parameters       %
%                                         %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% the maximum amplitude of the excitation
A0=0.3;
% the half-width of the wave envelope
sig=4.014;

% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```matlab
% %                                        %
% %          Elasticity Parameters         %
% %                                        %
% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% % lame constant
% lambda=40.38*10^-5;
% % the displacement field in
% u_x=zeros(Nx*Ny,2);
%
%
% % let's generate the grid
% for i=1:Nx
%     for j=1:Ny
%         % the counter
%         count=(i-1)*Ny+j;
%         x_grid(count,1)=x_mother(i);
%         x_grid(count,2)=y_mother(j);
%     end
% end
%
%
% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% %                                        %
% %     Generated Displacement Field       %
% %                                        %
% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% [ u_x, k_ny, k_nx, alpha_n ] = PrimaryDispField( ...
    omega,lamz,D,A0,L,H,Nx,Ny );
% % tumor itself
% [ f ] = getTumor( A,sig,x_0,L,Nx );
  [ f ] = getTumorAgnesi( A,sig,x_0,L,Nx );
% % disturbance due to tumor
% [ u_xt ] = getTumorDisturbace( A,f,u_x,L,H,omega,Nx,Ny);

% the combined function
[u_x,res,x_grid] = mainTumor( A,A0,sig,lamz,D,L,H,omega,Nx,Ny,x_0);
%% let's reshape vectors to plot them
U_r = reshape(u_x(:,1),Ny,Nx);
V_r = reshape(u_x(:,2),Ny,Nx);
% Ut_r = reshape(u_xt(:,1),Ny,Nx);
% Vt_r = reshape(u_xt(:,2),Ny,Nx);
Ut_r = reshape(res(:,1),Ny,Nx);
Vt_r = reshape(res(:,2),Ny,Nx);
x_r = reshape(x_grid(:,1),Ny,Nx);
y_r = reshape(x_grid(:,2),Ny,Nx);

%% let's generate the countor plot
figure(1)

contourf(x_r/lamz,y_r/lamz,U_r/A0,'LineColor','none')
colorbar
title('The displacement in x direction')
xlabel('x/\lambda_z')
ylabel('y/\lambda_z')
hcb=colorbar
title(hcb,'u_{x}/A_0')
```

```

%% let's generate the countor plot
figure(2)

contourf(x_r/lamz,y_r/lamz,V_r/A0,'LineColor','none')
colorbar
title('The displacement in y direction')
xlabel('x/\lambda_z')
ylabel('y/\lambda_z')
hcb=colorbar
title(hcb,'u_{y}/A_0')

% figure(3)
% plot(alpha_n)
% title('Fourier Coefficients')
% xlabel('n')
% ylabel('\lambda_n')

figure(4)

contourf(x_r/lamz,y_r/lamz,Ut_r/A0,'LineColor','none')
colorbar
title('The displacement due to tumor in x direction')
xlabel('x/\lambda_z')
ylabel('y/\lambda_z')
hcb=colorbar
title(hcb,'u_{x}/A_0')

%% let's generate the countor plot
figure(5)

contourf(x_r/lamz,y_r/lamz,Vt_r/A0,'LineColor','none')
colorbar
title('The displacement due to tumor in y direction')
xlabel('x/\lambda_z')
ylabel('y/\lambda_z')
hcb=colorbar
title(hcb,'u_{y}/A_0')

% tumor disturbance
figure(6)
plot(x_mother,f)
```

```
function [u_x,res,x_grid] = mainTumor( ...
    A,A0,sig,lamz,D,L,H,omega,Nx,Ny,x_0)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                                          %
%        Wave Energetics Parameters        %
%                                          %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% the maximum amplitude of the excitation
%A0
% the half-width of the wave envelope
%sig
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                                         %
%          Excitation Parameters          %
%                                         %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% the freqency of the excitation
%omega
% the wave length in z direction
%lamz
% the wave number in z direction
kz=2*pi/lamz;
% the tumor amplitude
%A
% the distance between the excitation regions
%D

% the mother interval
x_mother=linspace(0,L,Nx);
y_mother=linspace(-H/2,H/2,Ny);


%% the grid
x_grid=zeros(Nx*Ny,2);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                                         %
%        Wave Energetics Parameters       %
%                                         %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% the maximum amplitude of the excitation
%A0

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                                         %
%          Elasticity Parameters          %
%                                         %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% lame constant
lambda=40.38*10^-5;

% let's generate the grid
for i=1:Nx
    for j=1:Ny
        % the counter
        count=(i-1)*Ny+j;
        x_grid(count,1)=x_mother(i);
        x_grid(count,2)=y_mother(j);
    end
end


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                                         %
%     Generated Displacement Field        %
```

```
%                                         %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

[ u_x, k_ny, k_nx, alpha_n ] = PrimaryDispField( ...
    omega,lamz,D,A0,L,H,Nx,Ny );
% tumor itself
[ f ] = getTumor( A,sig,x_0,L,Nx );
% disturbance due to tumor
[ u_xt ] = getTumorDisturbace( A,f,u_x,L,H,omega,Nx,Ny);
%% let's reshape vectors to plot them
% U_r = reshape(u_x(:,1),Ny,Nx);
% V_r = reshape(u_x(:,2),Ny,Nx);
% Ut_r = reshape(u_xt(:,1),Ny,Nx);
% Vt_r = reshape(u_xt(:,2),Ny,Nx);
% x_r = reshape(x_grid(:,1),Ny,Nx);
% y_r = reshape(x_grid(:,2),Ny,Nx);


res = (u_x+u_xt)/A0;
end
```

```
function [ u_x, k_ny, k_nx, alpha_n ] = PrimaryDispField( ...
    omega,lamz,D,A,L,H,Nx,Ny )
%this function computes the primary displacement field

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                                         %
%          Excitation Parameters          %
%                                         %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% time: the time of the plot
% omega  the freqency of the excitation
% lamz: the wave length in z direction
% the excitaion amplitude
% D:the distance between the excitation regions
% the wave number in z direction
kz=2*pi/lamz;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                                         %
%             Grid Parameters             %
%                                         %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% the number of points in x direction
% Nx
% the number of points in y direction
% Ny
% the length of the domain
%L
%H

% the mother interval
x_mother=linspace(0,L,Nx);
y_mother=linspace(-H/2,H/2,Ny);

```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                                         %
%          Elasticity Parameters          %
%                                         %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% lame constant
lambda=40.38*10^-5;
% the displacement field in
u_x=zeros(Nx*Ny,2);


% let's generate the grid
for i=1:Nx
    for j=1:Ny
        % the counter
        count=(i-1)*Ny+j;
        x_grid(count,1)=x_mother(i);
        x_grid(count,2)=y_mother(j);
    end
end


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                                         %
%     Generated Displacement Field        %
%                                         %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% modes
ny=12;
% the wave number in y direction
for i=1:ny
    k_ny(i,1)=getArgY(i,H );
    % the wave number in x direction
    k_nx(i,1)=(omega^2/lambda-k_ny(i,1)^2)^0.5;
    % fourier coefficients
    alpha_n(i,1)= getFourierCoeff( k_nx(i,1),k_ny(i,1),kz,H,D,A,L);
end

% let's perform the displacement field computation
for j=1:ny
    for i=1:Nx*Ny
        % the argument of cosine in y direction
        phi_y=k_ny(j,1)*x_grid(i,2);
        % the argument in x direction
        phi_x=k_nx(j,1)*x_grid(i,1)-k_nx(j,1)*L+pi/2;
        % let's multiply amplitude with cos(phi_x)*sin(phi_y)
        u_x(i,1)=u_x(i,1)+alpha_n(j,1)*cos(phi_x)*sin(phi_y);
        u_x(i,2)=u_x(i,2)+(k_ny(j,1)/k_nx(j,1))*alpha_n(j,1)*sin(phi_x)*sin(phi_y);
    end
end

end
```

```
function [ k_y ] = getArgY( n,H )
%this function returns nth argument in y direction
```

```
% n is the order of the argument in y direction
% y is the coordinate
% H  is the domain length in y direction
k_y=(pi/H +2*(n-1)*pi/H);

end
```

```
function [ alpha ] = getFourierCoeff(knx,kny,ky,H,D,A,L)
%this function calculates fourier coefficients
% kny is the wave number of each mode
% ky is the wave number of excitation
% D is the distance between each exciation source
% A is the excitation amplitude
% H is the domain height
% L is the domain length
% the wave length of excitation
lambda=2*pi/ky;
% the fourier coefficients
alpha=cos((kny+ky)*(D-lambda/2))-cos((kny+ky)*(D+lambda/2))+cos((kny-ky)*(D+lambda/2));
alpha=alpha-cos((kny-ky)*(D-lambda/2));
alpha=(-2.*A/L)*alpha/cos(pi/2-knx*L);
end
```

```
function [ f ] = getTumor( A,sig,x_0,L,Nx )
% this function returns the shape of the tumor

% the mother interval
x=linspace(0,L,Nx);

% function itself
for i=1:Nx
    f(i,1)=A*(x(i)-x_0)*exp(0.5*((x(i)-x_0)/(sig))^2)/sig^2;
   %  f(i,1)=A*exp(0.5*((x(i)-x_0)/(sig))^2);
end

end
```

```
function [ f ] = getTumorAgnesi( A,sig,x_0,L,Nx )
% this function returns the shape of the tumor

% the mother interval
x=linspace(0,L,Nx);
    % function itself
    for i=1:Nx
        f(i,1)=-2.*A*(x(i)-x_0)/((1+(sig*(x(i)-x_0))^2)^2);
    end

end
```

```
function [ u_xt ] = getTumorDisturbace(A,f,u_x,L,H,omega,Nx,Ny)
```

```matlab
% this function calculates the disturbance due to tumor
%f: tumor disturbance
% u_x: displacement field

% the mother interval
x_mother=linspace(0,L,Nx);
y_mother=linspace(-H/2,H/2,Ny);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                                          %
%          Elasticity Parameters           %
%                                          %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% lame constant
lambda=40.38*10^-5;
% modes
ny=12;
% alpha coefficients
alpha_n=zeros(ny,1);
for i=1:ny
    k_nx(i,1)=getArgY(i,H );
    % the wave number in x direction
    k_ny(i,1)=(omega^2/lambda-k_nx(i,1)^2)^0.5;
    % fourier coefficients
    for j=1:Nx
        alpha_n(i,1)=  ...
            alpha_n(i,1)-2.*sin(k_nx(i,1)*x_mother(1,j))*f(j,1)*u_x(j,1);
    end
end

% the displacement field in
u_xt=zeros(Nx*Ny,2);

% let's generate the grid
for i=1:Nx
    for j=1:Ny
        % the counter
        count=(i-1)*Ny+j;
        x_grid(count,1)=x_mother(i);
        x_grid(count,2)=y_mother(j);
    end
end

for j=1:ny
    for i=1:Nx*Ny
        % the argument of cosine in y direction
        phi_y=k_ny(j,1)*x_grid(i,2);
        % the argument in x direction
        phi_x=k_nx(j,1)*x_grid(i,1)-k_nx(j,1)*L+pi/2;
        % let's multiply amplitude with cos(phi_x)*sin(phi_y)
        u_xt(i,1)=u_xt(i,1)+alpha_n(j,1)*cos(phi_x)*sin(phi_y);
        u_xt(i,2)=u_xt(i,2)+(k_ny(j,1)/k_nx(j,1))*alpha_n(j,1)*sin(phi_x)*sin(phi_y);
    end
end
% let's multiply it with amplitude
u_xt=A*u_xt;
```

```
end
```

Appendix B

Weak Formulation: Mass Matrix and Stiffness Matrix

As a first example of weak formulation, heat equation in 1-D can be used as it has diffusive property:

$$u_t = u_{xx}. \tag{B.1}$$

Moreover, u can be written in terms of Legendre polynomials as follows:

$$u = \sum_{i=1}^{N_x} \alpha_i(t) P_i(x) \tag{B.2}$$

where α_i is the corresponding coefficient of a Legendre polynomial and N_x is the order of Legendre polynomials used. If the spectral decomposition B.2 is replaced back into equation B.1, the following equation is obtained:

$$\sum_{i=1}^{N_x} \dot{\alpha}_i P_i = \sum_{i=1}^{N_x} \alpha_i P_i'' \tag{B.3}$$

If the equation B.3 is multiplied with P_j which is a Legendre polynomial at j_{th} order and integrated over whole domain, the following form equation is obtained:

$$\sum_{j=1}^{N_x} \sum_{i=1}^{N_x} \dot{\alpha}_i P_i P_j w_i = \sum_{j=1}^{N_x} \sum_{i=1}^{N_x} \alpha_i P_i'' P_j w_i. \tag{B.4}$$

By the functional orthogonality of Legendre polynomials, left hand side of equation B.4 reduces to the mass matrix. Moreover, using integration by part, right hand side of equation can also be represented in terms of first derivative products of Legendre polynomials. As a result, the following equation is obtained:

$$\dot{\alpha}_j w_j = -\sum_{i=1}^{N_x} \alpha_i P_i' P_j' w_i \tag{B.5}$$

In matrix form, equation B.5 can be given as:

$$[\mathbf{M}]\dot{\alpha} = [\mathbf{K}]\alpha. \tag{B.6}$$

where $[\mathbf{M}]$ is a diagonal mass matrix with diagonal entries w_j and $[\mathbf{K}]$ is the stiffness matrix which is a full matrix.

Bibliography

[1] S. Budday, G. Sommer, C. Birkl, C. Langkammer, J. Haybaeck, J. Kohnert, M. Bauer, F. Paulsen, P. Steinmann, E. Kuhl, and G. A. Holzapfel. Mechanical characterization of human brain tissue. *Acta Biomaterialia*, 48(October):319–340, 2017.

[2] Silvia Budday, Paul Steinmann, and Ellen Kuhl. The role of mechanics during brain development. *Journal of the Mechanics and Physics of Solids*, 72:75–92, 2014.

[3] E. H. Clayton, G. M. Genin, and P. V. Bayly. Transmission, attenuation and reflection of shear waves in the human brain. *Journal of The Royal Society Interface*, 9(76):2899–2910, 2012.