

ViscoWave Documentation and User Guide

by:

Hyung S. Lee, Ph.D., P.E.



100 Trade Centre Dr., Suite 200

Champaign, Illinois 61820

September 2022

Table of Contents

CHAPTER 1 – INTRODUCTION	1
Falling Weight Deflectometer.....	1
Backcalculation General	3
Dynamic Backcalculation	3
ViscoWave – A New Finite Layer Solution	5
CHAPTER 2 – Objective of Open-Source ViscoWave.....	9
Limitations of Current Backcalculation Procedures	9
Purpose and Limitations of Original ViscoWave	10
Objective and Scope of Open-Source ViscoWave	11
CHAPTER 3 – ViscoWave Formulation.....	12
Introduction.....	12
Summary of ViscoWave Formulation	12
Fundamental Equations.....	12
Element Stiffness Matrix	14
Global Stiffness Matrix.....	17
Boundary Conditions for a Circular Unit Impulse Loading at the Ground Surface	19
Numerical Inversion of Laplace and Hankel Transforms.....	20
System Response To Arbitrary Loading.....	22
Interpolation of Impulse Response Using Cubic Spline	22
Material Models (Elastic and Viscoelastic)	24
Interconversion between Relaxation Modulus and Dynamic Modulus.....	30
Conversion from Relaxation Modulus (Time Dependent) to Dynamic Modulus (Frequency Dependent).....	34
Conversion from Dynamic Modulus (Frequency Dependent) to Relaxation Modulus (Time Dependent).....	35
Time-Temperature Superposition and Modulus Master Curve	37
CHAPTER 4 – ViscoWave User Guide	40
License Agreement	40
Installing ViscoWave and Portable R	40
Before Getting Started	43
Getting Started	45

Viscowave Analysis Menu Options.....	45
Pre-Processing of FWD Data.....	46
Extracting Time History Data from Dynatest FWD Files	47
Extracting Time History Data from JILS FWD Files.....	47
Extracting Time History Data from Kuab FWD Files.....	48
File Structure for Extracted Time History Data.....	50
Forward Simulation Using Viscowave	53
Backcalculation Using Viscowave (Single Drop)	55
Converting Relaxation Modulus to Dynamic Modulus	57
Backcalculation Using Batch Mode.....	58
Post-Processing for Batch Mode Backcalculation.....	60
REFERENCES	65

CHAPTER 1 – INTRODUCTION

FALLING WEIGHT DEFLECTOMETER

Falling Weight Deflectometer (FWD) is one of the most frequently used nondestructive testing (NDT) devices for evaluating the structural integrity of an existing pavement. As its full name implies, the FWD is equipped with a falling mass mechanism capable of inducing an impact load on the pavement surface. Due to the nature of the impact load generated by a falling mass, the load typically has a short duration (usually 20 ms to 40 ms) and generates a stress wave that propagates through the pavement structure. The resulting time-dependent response of the pavement structure or more specifically, the vertical deflection at the pavement surface resulting from the stress wave is measured at various radial distances from the load and is recorded for the structural analysis of the pavement system.

There are three primary FWD models being used in the U.S., namely the Dynatest model (Figure 1), JILS model (Figure 2), and Kuab model (Figure 3). These FWDs have inherent differences in terms of their overall design, hardware, and software. However, all these FWDs operate under the same principle described above.



Figure 1. Falling Weight Deflectometer, Dynatest Model (Courtesy Dynatest.com)

The FWD results are mainly used for assessing the structural characteristics of the pavement system as a whole, or the individual layers that make up the pavement. Different analysis procedures are available and have been used in the past for this purpose, including those that are very simple (but efficient to use) such as the Deflection Basin Parameters and those based on Boussinesq's equation for a halfspace. However, the focus of this manual is on one of the more

advanced procedures called “Backcalculation” which will be discussed further in the following sections.



Figure 2. Falling Weight Deflectometer, JILS Model



**Figure 3. Falling Weight Deflectometer, Kuab Model (Courtesy
<https://ceer.iastate.edu/kuab-falling-weight-deflectometer/>)**

BACKCALCULATION GENERAL

The process of estimating material parameters from the FWD data can be categorized as an inverse problem whose objective is to determine the system characteristics (e.g., layer modulus) from the known input (e.g., applied load) and output (e.g., measured deflection). In the pavement engineering community, such inverse problems have typically been solved using a procedure commonly referred to as backcalculation.

In general, backcalculation of layer parameters is carried out by matching the FWD load and deflection to those from a theoretical model. Therefore, as is the case for most inverse problems, the two crucial components of a backcalculation methodology are:

1. A **forward solution** or a theoretical, mechanistic model capable of simulating the FWD load and deflection.
2. An **iterative or statistical routine** capable of determining the optimum layer parameters that minimize the error between the measured and simulated results.

Numerous combinations of the forward and iterative solutions have been developed and used to date for layer modulus backcalculation. As such, it is close to impossible to list the characteristics of all existing backcalculation methodologies.

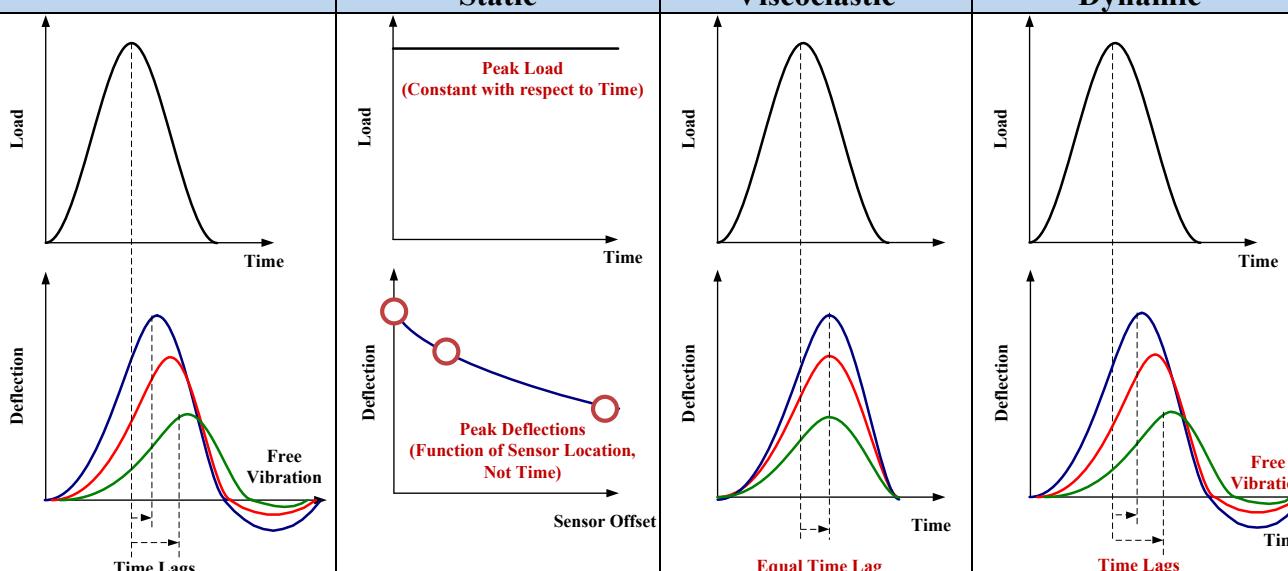
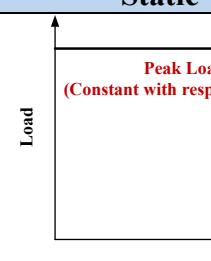
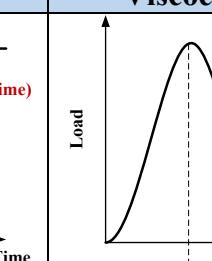
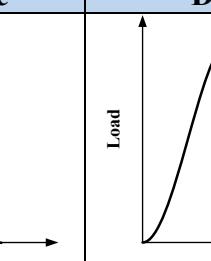
However, it is worthwhile to note that the existing procedures can be classified into three general categories depending on the assumptions and the characteristics of the forward solution – static, viscoelastic, and dynamic backcalculation methods.

DYNAMIC BACKCALCULATION

The characteristics of the FWD deflections as well as the three mechanistic models are summarized in Table 1. The forward solutions used for **static backcalculation** procedures assume that the pavement structure under the applied load is in static equilibrium (i.e., time-independent). As a result, these models do not allow for incorporating the time-dependent behavior of the material. Although the static models are extremely efficient, these solutions do not account for the dynamic nature (time-dependent, impact) of the FWD load and deflections.

As it has been proven by numerous researchers, Asphalt Concrete (AC) that comprises the top layer of flexible pavements is viscoelastic in nature. Unlike an elastic material, viscoelastic materials show time-dependent response even under a static (or constant) load. However, as it was pointed out by Kutay et. al. (2011), the **viscoelastic solutions** do not consider the effect of wave propagation attributed to the impact load. As such, they are not capable of modeling the different time delays measured at different sensors nor the free vibration of the pavement structure, which may be significant depending on the pavement structure and its surrounding environmental conditions.

Table 1. Characteristics of FWD time histories and mechanistic models (static, viscoelastic, and dynamic)

Description	Measured FWD Data	Type of Mechanistic Model (Forward Solution)		
		Static	Viscoelastic	Dynamic
Schematics		 Peak Load (Constant with respect to Time)	 Peak Deflections (Function of Sensor Location, Not Time)	 Equal Time Lag
Model Efficiency	Not applicable	Extremely efficient	Efficient	Inefficient
Material Time-Dependency	Present due to viscoelastic AC	Not considered	Considered	Considered
Inertial, Dynamic Effects	Present due to impact load	Not considered	Not considered	Considered
Deflection Time Lags	Different for each sensor	Not considered	Equal time lag for all sensors	Considered (Different for each sensor)
Free Vibration	Present depending on pavement structure	Not considered	Not considered	Considered

The forward models adopted in the **dynamic backcalculation** methodologies are typically based on solutions that stem from the theory of elastic wave propagation. Therefore, the stress wave propagation or the time-dependent inertial effect is naturally modeled in these solutions. The viscoelasticity of the material typically has been taken into consideration through the concept of a damping ratio derived from the theory of vibrations (Chatti and Yun, 1996; Chatti et. al., 2004; Ji, 2005; Matsui et. al., 2011) or through the elastic-viscoelastic correspondence principle (Christensen, 2003; Findley et. al., 1976; Kim, 2009). As such, the dynamic solutions provide realistic pavement responses that are in better agreement with the measured responses.

Most of the recent developments in dynamic backcalculation were made with mechanistic models known as the Finite Layer Method (FLM). As schematically shown in Figure 4, FLM is similar to the Finite Element Method (FEM) in the sense that the geometry of the entire structure (i.e., a flexible pavement) is broken down into pieces or elements for the analysis. The difference is in the mathematical formulation of the analysis elements and the way the elements make up the pavement structure.

The FEM requires each layer of the pavement be divided into a large number of smaller elements to ensure accurate and reliable results. Due to the large number of elements, FEM usually involves a long computation time, which makes it inadequate for backcalculation. On the other hand, the fundamental concept behind the FLM is to define an entire layer (or sublayer) as a single element; thus for a 3-layer pavement system, FLM only needs 3 elements. This is made possible by taking advantage of the simplified geometry of each pavement layer (Figure 4b) and solving the wave equations analytically (or semi-analytically). In other words, the phenomenon of stress wave propagation within a given layer is solved in closed form (or semi-closed form). The interaction between multiple layers (or elements in this case) and the boundary conditions are then accounted for in the same manner as the FEM.

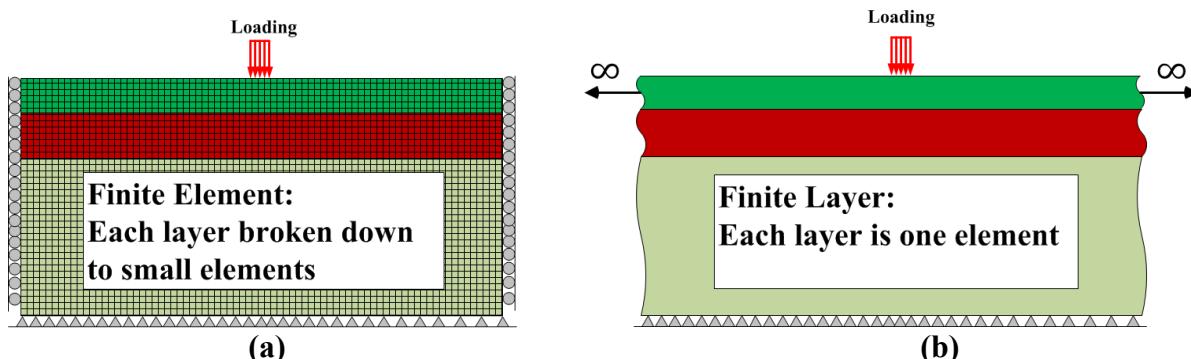


Figure 4. Conceptual schematics of (a) Finite Element and (b) Finite Layer methods

VISCOWAVE – A NEW FINITE LAYER SOLUTION

A common characteristic of the traditional FLM solutions is that the analytical developments for solving the wave equations were made in the frequency domain through the use of Discrete Fourier transform (DFT) for converting the load and deflection signals from time domain to

frequency domain. Figure 5 conceptually shows how the DFT is utilized in traditional FLM solutions. The three major steps of the traditional FLM can be described as the following:

1. Use the DFT to decompose the load signal into multiple, everlasting sinusoids (sine and cosine functions) of different amplitude and frequency.
2. For each sinusoidal loading, obtain the corresponding deflection which is also a sinusoid at the same frequency as the loading.
3. Sum all the sinusoidal deflections to obtain the deflection corresponding to the load signal.

The traditional FLM solutions utilizing the DFT showed varying levels of success in dynamic backcalculation using the theoretical FWD time histories generated by the forward solution. However, backcalculation using the field FWD data has generally shown fair to poor levels of success (Chatti and Yun, 1996; Chatti et. al., 2004; Ji, 2005; Matsui et. al., 2011; Al-Khoury et. al., 2001a, 2001b, 2002). It has been recognized that the difficulties in dynamic backcalculation arise primarily from the inevitable use of DFT on truncated signals (Chatti et. al., 2004; Ji, 2005).

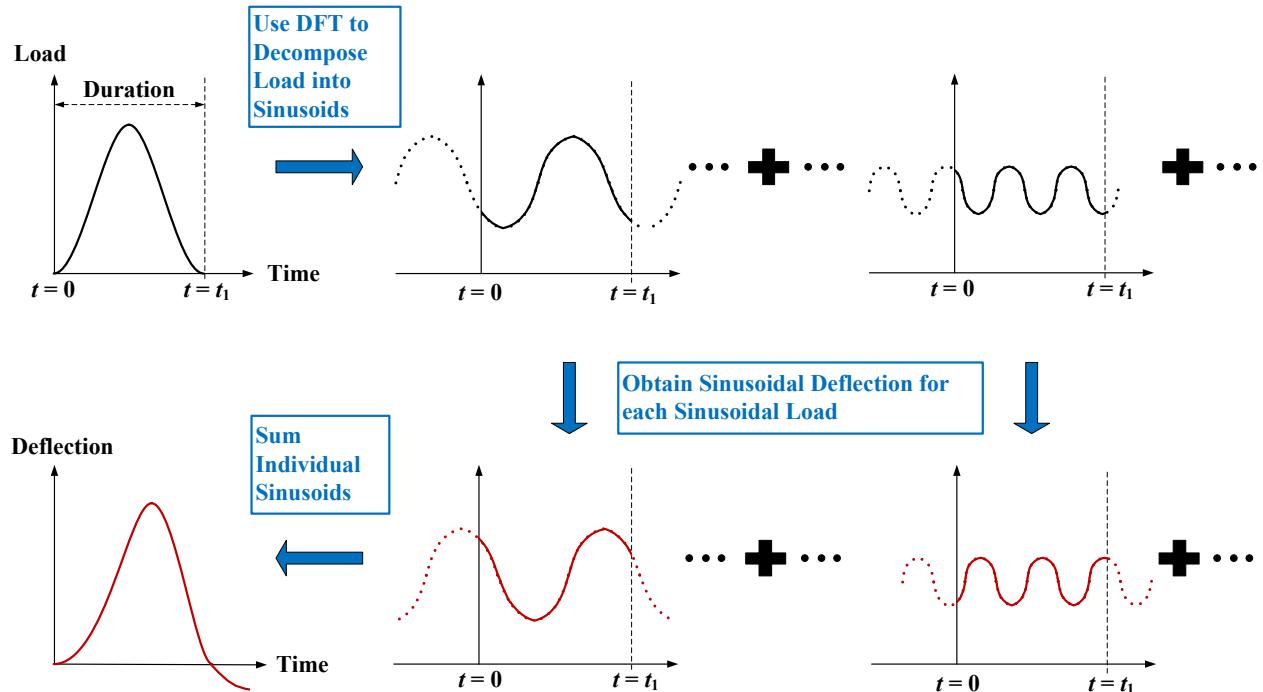


Figure 5. Conceptual schematics of traditional FLM utilizing DFT

As an example, Figure 6 shows typical FWD deflection time histories extracted from the Long Term Pavement Performance (LTPP) database. Figure 6a shows the deflection vibrating freely after the end of the loading duration while Figure 6b shows the deflection recovering slowly after the load. Regardless of the cause, both time histories in Figure 6 show that the deflections at the end of the time window are not at rest due to signal truncation (i.e., the recording of the time history was stopped before the deflections came back to zero). The truncated signal leaves a discontinuity in the deflection time history, which causes significant challenges to the DFT. Theoretically speaking, because of the everlasting sinusoids that form the basis of signal

decomposition in DFT (see Figure 5), it requires an infinite number of sinusoids to accurately model such a rapid change or discontinuity in the signal. While the issue of signal truncation may be alleviated by using a large number of sinusoids, such an approach is associated with a significant loss of computational efficiency.

In addition to the above, dynamic backcalculation – especially with field FWD data – has further been challenged by the periodicity assumption in the DFT algorithm not being able to accurately disclose the frequency content of the transient FWD time histories and the DFT being very sensitive to noise in the FWD data (Chatti et. al., 2004; Ji, 2005; Matsui et. al., 2011).

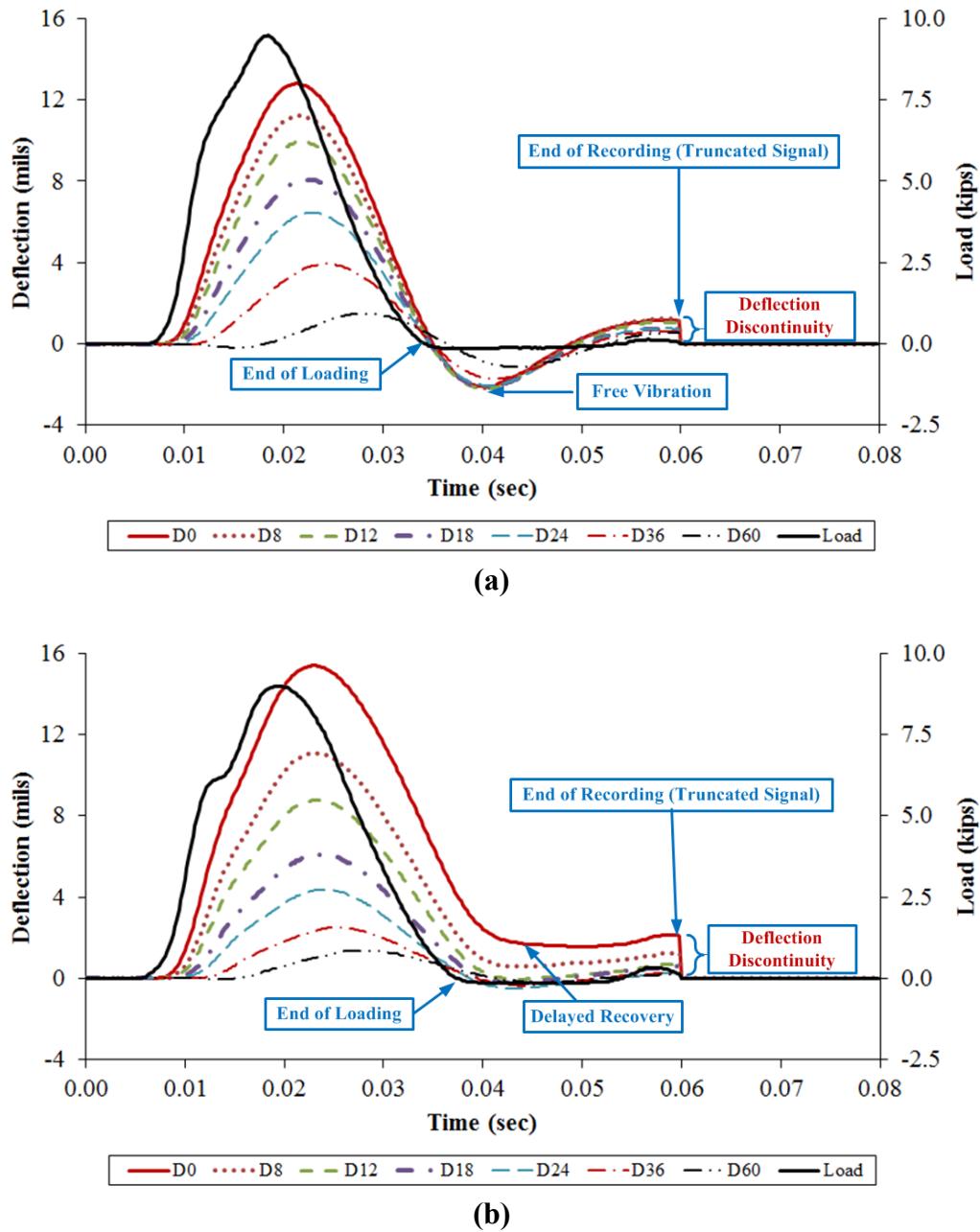


Figure 6. Typical FWD deflection time histories (a) with and (b) without free vibration

More recently, Lee (2013, 2014) introduced a new FLM solution called ViscoWave which was derived using continuous transforms in place of the discrete transforms such as the DFT. The primary advantage of using the continuous transforms is that it allows for the load signal to be decomposed into impulses rather than sinusoids. As schematically shown in Figure 7, the response to each impulse load is obtained and then summed to yield the final response. Because ViscoWave solution is based on impulse loads and impulse responses, it is more adequate for transient, non-periodic signals and is not affected by signal truncation.

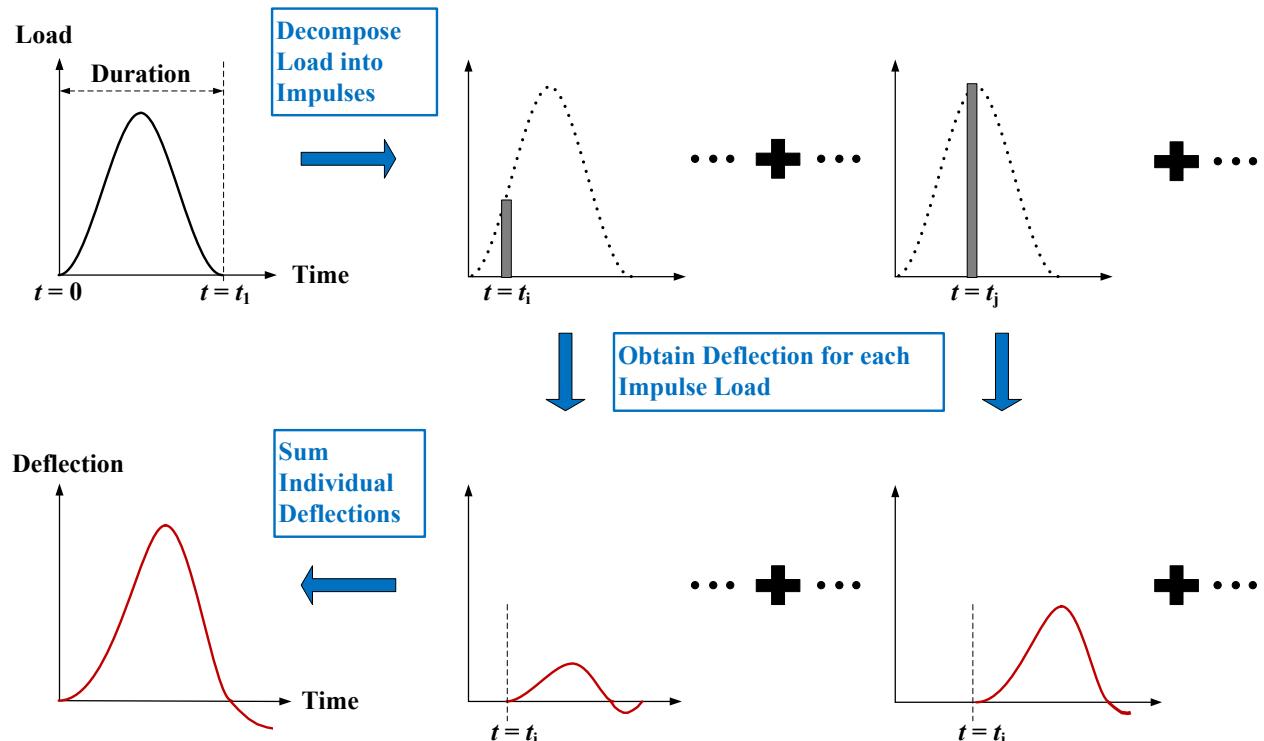


Figure 7. Conceptual schematics of ViscoWave utilizing impulse loads and responses

A unique characteristic of ViscoWave is that it uses **impulse loads** and **impulse responses**.

Furthermore, because the impulse response is a fundamental response function, ViscoWave can also be used for obtaining the pavement response under a moving load. As such, the ViscoWave algorithm has recently been enhanced for moving load simulation and at the same time, for backcalculating the layer moduli from the deflection measured under a moving load. However, the moving load simulation (and backcalculation) is beyond the scope of the released version of ViscoWave, and hence this manual.

CHAPTER 2 – OBJECTIVE OF OPEN-SOURCE VISCOWAVE

LIMITATIONS OF CURRENT BACKCALCULATION PROCEDURES

At the project level, deflection analysis or backcalculation is conducted for a variety of purposes such as pavement design, analysis, forensic investigation, and pavement management. However, the discussion herein will be limited to the use of backcalculation in pavement design.

With the development and implementation of various mechanistic-empirical (ME) pavement design procedures in recent years, significant improvements have been made to pavement design/analysis methodologies. The Mechanistic-Empirical Pavement Design Guide (MEPDG), developed under NCHRP Project 1-37A, represents the state of the practice and is the most frequently used ME-based method for designing pavements (ARA, 2004).

Similar to any other design methodologies, the first step in MEPDG rehabilitation design of an existing flexible pavement is to characterize the AC layer in place. The MEPDG characterizes the existing AC layer by a “damaged master curve” which becomes the basis for calculating future AC responses and fatigue damage in the existing AC layers after the placement of a new overlay. In the MEPDG input Level 1 procedure, the damaged master curve is constructed based on the backcalculated modulus of the AC layer (i.e., static backcalculation). As schematically shown in Figure 8, the Level 1 procedure involves the following:

1. Backcalculate the AC modulus from FWD deflection data (Denoted as E_{FWD} in Figure 8).
2. Construct the undamaged AC dynamic modulus master curve from available predictive equations or from laboratory testing of the AC mixture.
3. From the master curve, estimate the undamaged AC dynamic modulus, E_{PRED} , corresponding to the temperature and time (or frequency) of E_{FWD} .
4. Calculate AC damage, d_j , using the equation shown in Figure 8.
5. Construct the damaged master curve by shifting the undamaged master curve down such that it passes through E_{FWD} .

The above procedure, however, is challenged by the use of static backcalculaion methods. As mentioned in the previous chapter of this manual, time- or frequency-dependent information cannot be modeled using these (static) solutions. Consequently, static backcalculation yields a single modulus value (E_{FWD}), rather than the entire time-dependent modulus curve of a viscoelastic material.

Furthermore, E_{FWD} is independent of time or frequency. As a result, the frequency corresponding to E_{FWD} has traditionally been assumed to be a constant (typically between 15 Hz to 30 Hz) or has been estimated through various empirical equations as a function of the FWD loading duration (Al-Qadi et. al., 2008).

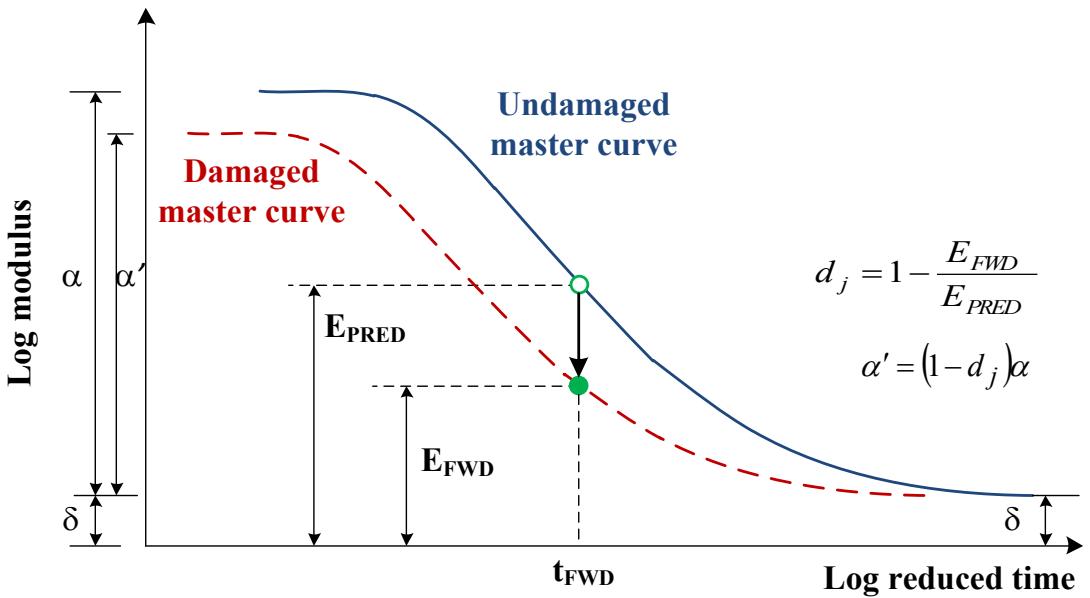


Figure 8. Current MEPDG Level 1 procedure for constructing modulus master curve for existing AC

PURPOSE AND LIMITATIONS OF ORIGINAL VISCOWAVE

ViscoWave was originally developed to provide the pavement engineering community with a tool that can be used for dynamic backcalculation of FWD data. However, the original version of ViscoWave developed by Lee (2013) had several shortcomings described below.

1. ViscoWave was not fast enough for simulating the FWD time history, let alone backcalculation. The following provides a brief history.
 - a. ViscoWave routine that was written in MatLab took approximately 30 minutes for a single run. A single backcalculation (which requires repeated runs of ViscoWave) took approximately a week.
 - b. ViscoWave routine that was translated into C++ environment took approximately 30 seconds for a single run and approximately a day for backcalculation.
 - c. The above routine was upgraded such that ViscoWave implemented multi-core processing (i.e., it allowed to use all CPU cores available in the machine). This reduced the ViscoWave runtime to approximately 1 second and approximately 5 minutes for backcalculation.
 - d. A cubic-spline interpolation scheme was implemented to further reduce the runtime. Implementation of cubic-spline reduced the ViscoWave runtime to approximately 0.1 second or less, and approximately 1 to 3 minutes for backcalculation (depending on the problem – using good seed values help reduce the runtime).
2. ViscoWave did not have a good user interface. While a temporary interface in MS Excel was available, it did not provide enough flexibility and functionality to be shared with the community. E.g., there were so many ways that could cause the interface to crash and setting up the interface for backcalculation was a time-consuming effort of its own.

3. Lack of a detailed manual along with upgrades that were never documented or published. E.g., while the cubic-spline interpolation discussed previously allowed to reduce the ViscoWave runtime significantly, it has never been documented.

OBJECTIVE AND SCOPE OF OPEN-SOURCE VISCOWAVE

The primary objective of releasing ViscoWave as an open-source code is to provide the pavement engineering community with an easy-to-use tool for dynamic backcalculation of FWD data.

At the time of this writing, ViscoWave Version 3.0 is the latest version that is released in the open-source repository. This version of ViscoWave overcomes the limitations of the original ViscoWave discussed previously along with several other updates. More specifically, features of the open-source ViscoWave provides the following features.

1. The C++ code has been reviewed by a software engineer for readability and optimization of the code for runtime speed. In other words, the ViscoWave source code has been updated to meet the industry standards (or best practices) on C++ documented in: <https://google.github.io/styleguide/cppguide.html>.
2. A user-friendly interface has been developed and released for Version 3.0. The interface was developed in MS Excel and Visual Basic for Application (VBA) environment. The interface has its own code in VBA language. However, the VBA code was written only to help the users with several convenient features such as navigating through the menu options, plotting of the results, batch processing, etc. The core engine for ViscoWave is still in C++.
3. A user manual that documents the ViscoWave updates that had never been published and to provide a step-by-step guide for the user in using ViscoWave. This is the purpose of this manual.

CHAPTER 3 – VISCOWAVE FORMULATION

INTRODUCTION

The detailed mathematical development of the original ViscoWave has been published in the past by Lee (2013, 2014). Since then, some additional features have been developed and implemented into ViscoWave. However, the details of these additional features have never been made available to the public in the form of a document or publication.

As such, the objective of this chapter is twofold. The first objective is to briefly review the original formulation of ViscoWave. The second and the more important objective is to document the mathematical formulation of the additional features that have been implemented since 2014.

 If you **do not need the mathematical details** behind ViscoWave, you may skip this chapter completely. In such case, **please proceed to the next chapter**.

SUMMARY OF VISCOWAVE FORMULATION

First, it is necessary that the important aspects of ViscoWave formulation be reviewed. However, this review will remain as brief as possible and will focus on the developments that had not been documented previously. The readers interested in learning more details of original ViscoWave development are referred to Lee (2013, 2014).

Fundamental Equations

As discussed previously, a unique characteristic of the FLM is that the wave equations within an element (that is, the layer or sublayer) are solved analytically (or semi-analytically). Therefore, the formulation of ViscoWave starts with the classical equation of motion for a continuous medium given as the following.

$$\nabla \cdot \sigma + b = \rho \ddot{u} \quad (1)$$

where σ is the stress tensor, b is the vector of body forces per unit volume, ρ is the mass density of the material, and u is the displacement vector. Substituting the strain-displacement relationship from the theory of elasticity and the stress-strain relationship from the theory of viscoelasticity into the above equation yields the following wave equation for viscoelastic materials (Lee, 2013; Lee, 2014).

$$(\lambda + \mu) * \nabla (\nabla \cdot u) + \mu * \nabla^2 u = \rho \ddot{u} \quad (2)$$

where λ and μ are the lamé constants, the function $\alpha * \beta$ represents the well-known Stieltjes convolution integral defined as $\alpha * \beta = \int_0^t \alpha(t-\tau) \frac{\partial \beta(\tau)}{\partial \tau} d\tau$.

By means of Helmholtz decomposition and after adopting a cylindrical coordinate system as shown in Figure 9, the above wave equation can be decoupled into two independent equations in terms of scalar potentials.

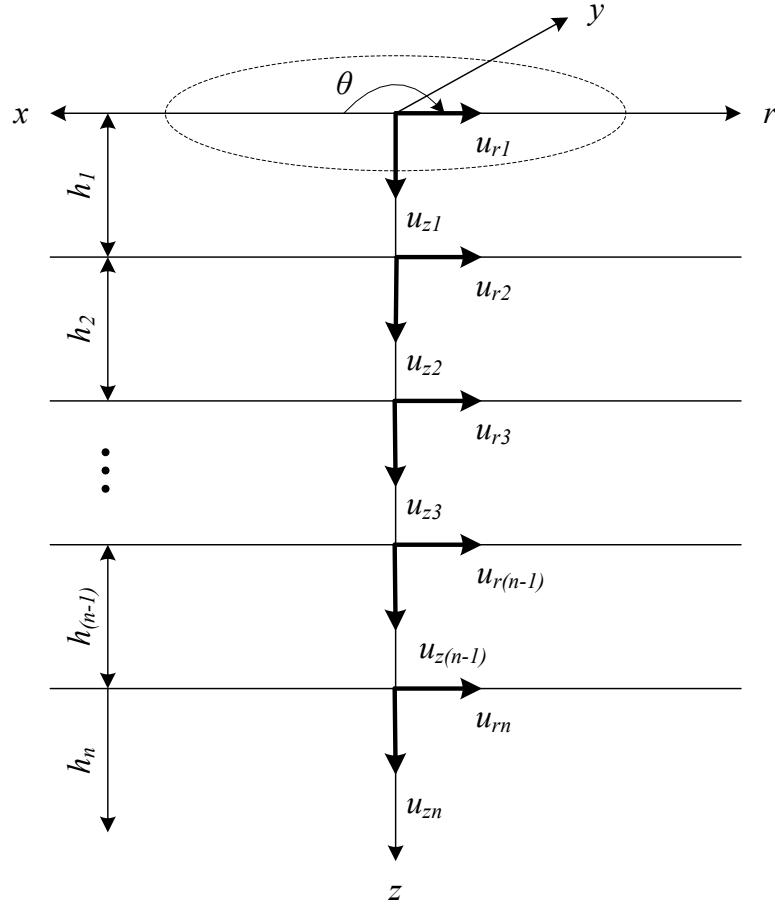


Figure 9. Coordinate System for Axisymmetric Layers on a Halfspace

In the Laplace-Hankel transformed domain, these decoupled wave equations are of the following forms.

$$\frac{\partial^2 \bar{\Phi}}{\partial z^2} - \left(k^2 + \frac{s}{\hat{c}_1^2} \right) \bar{\Phi} = 0 \quad (3)$$

$$\frac{\partial^2 \bar{\Psi}}{\partial z^2} - \left(k^2 + \frac{s}{\hat{c}_2^2} \right) \bar{\Psi} = 0 \quad (4)$$

where $\bar{\Phi}$ and $\bar{\Psi}$ are the Laplace-Hankel transforms of the scalar potentials Φ and Ψ , and s and k are the Laplace and Hankel variables, respectively. In addition, \hat{c}_1^2 and \hat{c}_2^2 in the above equations are given as:

$$\hat{c}_1^2 = \frac{\hat{\lambda} + 2\hat{\mu}}{\rho} \quad (5)$$

and

$$\hat{c}_2^2 = \frac{\hat{\mu}}{\rho} \quad (6)$$

where $\hat{\lambda}$ and $\hat{\mu}$ are the Laplace-transformed lamé constants. The solutions of Equations (3) and (4) in the Laplace-Hankel domain are obtained as the following.

$$\bar{\Phi} = Ae^{-zf(k,s)} + Be^{-(h-z)f(k,s)} \quad (7)$$

$$\bar{\Psi} = Ce^{-zg(k,s)} + De^{-(h-z)g(k,s)} \quad (8)$$

where A, B, C , and D are arbitrary constants, h is the thickness of a given layer, and:

$$f(k,s) = \sqrt{k^2 + \frac{s}{\hat{c}_1^2}} \quad (9)$$

$$g(k,s) = \sqrt{k^2 + \frac{s}{\hat{c}_2^2}} \quad (10)$$

Element Stiffness Matrix

To derive the stiffness matrix for a 2-noded element (i.e., layer with a finite thickness), the displacements and the relevant stresses at the upper and lower boundaries (i.e., at $z = 0$ and at $z = h$, respectively) need to be obtained from the potentials shown in Equations (7) and (8).

In matrix form, the equations for the resulting displacements are given as the following.

$$\begin{Bmatrix} \bar{u}_{r1} \\ \bar{u}_{z1} \\ \bar{u}_{r2} \\ \bar{u}_{z2} \end{Bmatrix} = \begin{bmatrix} -k & -ke^{-hf} & kg & -kge^{-hg} \\ -f & fe^{-hf} & k^2 & k^2 e^{-hg} \\ -ke^{-hf} & -k & kge^{-hg} & -kg \\ -fe^{-hf} & f & k^2 e^{-hg} & k^2 \end{bmatrix} \begin{Bmatrix} A \\ B \\ C \\ D \end{Bmatrix} = \mathbf{S}_1 \cdot \begin{Bmatrix} A \\ B \\ C \\ D \end{Bmatrix} \quad (11)$$

where \bar{u}_{r1} and \bar{u}_{z1} are the radial and vertical displacements at the upper boundary, respectively. Similarly, \bar{u}_{r2} and \bar{u}_{z2} are their displacement counterparts at the lower boundary.

Also in matrix form, the equations for the relevant stresses are obtained as the following.

$$\begin{Bmatrix} \bar{\sigma}_{rz1} \\ \bar{\sigma}_{z1} \\ \bar{\sigma}_{rz2} \\ \bar{\sigma}_{z2} \end{Bmatrix} = s\hat{\mu} \cdot \begin{bmatrix} 2f & -2fe^{-hf} & -K & -Ke^{-hg} \\ K & Ke^{-hf} & -2k^2 g & 2k^2 ge^{-hg} \\ 2fe^{-hf} & -2f & -Ke^{-hg} & -K \\ Ke^{-hf} & K & -2k^2 ge^{-hg} & 2k^2 g \end{bmatrix} \begin{Bmatrix} A \\ B \\ C \\ D \end{Bmatrix} = s\hat{\mu} \cdot \mathbf{S}_2 \cdot \begin{Bmatrix} A \\ B \\ C \\ D \end{Bmatrix} \quad (12)$$

where,

$$K = k^2 + g^2 \quad (13)$$

By inverting the 4-by-4 matrix \mathbf{S}_1 in Equation (11) and substituting it into Equation (12), one obtains the equations relating the displacements to the stresses without the arbitrary constants. Then, by calculating the surface tractions from these stress equations, one finally obtains the stiffness matrix for a 2-noded element. The stiffness matrix, $\mathbf{S}_{2\text{-noded}}$, is given as the following.

$$\mathbf{S}_{2\text{-noded}} = s\hat{\mu} \cdot \mathbf{N} \cdot \mathbf{S}_2 \cdot \mathbf{S}_1^{-1} \quad (14)$$

where

$$\mathbf{N} = \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (15)$$

Additional details regarding the derivation of the stiffness matrix can be found in Lee (2013, 2014).

 The following (or more specifically **Equation (16) through Equation (37)**), provides the **stiffness matrix equations** for both the 2-noded and 1-noded elements in the Laplace-Hankel domain. These equations were **not documented previously** by Lee (2013, 2014).

In the original implementation of ViscoWave (Lee, 2013 & 2014), the 4 -by-4 matrix \mathbf{S}_1 in Equation (11) and in Equation (14) was inverted numerically. However, for the current

implementation of ViscoWave, the stiffness matrix (Equation (14)) has been derived entirely in closed form (i.e., the numerical inversion of \mathbf{S}_1 and the numerical matrix multiplication in Equation (14) have been eliminated) for efficiency and stability of the program.

The closed form stiffness matrix for a 2-noded element, $\mathbf{S}_{2\text{-noded}}$, is given as the following.

$$\mathbf{S}_{2\text{-noded}} = \frac{s\hat{\mu}}{D} \begin{bmatrix} S_{11} & S_{12} & S_{13} & S_{14} \\ & S_{22} & S_{23} & S_{24} \\ & & S_{33} & S_{34} \\ Sym & & & S_{44} \end{bmatrix} \quad (16)$$

The elements of the above stiffness matrix (S_{11} through S_{44}) are given as the following.

$$S_{11} = -fK_1(Q_1Q_4fg - Q_2Q_3k^2) \quad (17)$$

$$S_{12} = -k \left\{ -Q_1Q_2 \left(2f^2g^2 + k^2K \right) + fg \left(Q_3Q_4 - 4e^{-hf}e^{-hg} \right) \left(g^2 + 3k^2 \right) \right\} \quad (18)$$

$$S_{13} = 2fK_1 \left(Q_1e^{-hg}fg - e^{-hf}Q_2k^2 \right) \quad (19)$$

$$S_{14} = -2 \left(e^{-hf} - e^{-hg} \right) \left(1 - e^{-hf}e^{-hg} \right) f g k K_1 \quad (20)$$

$$S_{22} = -gK_1(Q_2Q_3fg - Q_1Q_4k^2) \quad (21)$$

$$S_{23} = 2 \left(e^{-hf} - e^{-hg} \right) \left(1 - e^{-hf}e^{-hg} \right) f g k \left(k^2 - g^2 \right) \quad (22)$$

$$S_{24} = -2gK_1 \left(-e^{-hf}Q_2fg + Q_1e^{-hg}k^2 \right) \quad (23)$$

$$S_{33} = -fK_1(Q_1Q_4fg - Q_2Q_3k^2) \quad (24)$$

$$S_{34} = k \left\{ -Q_1Q_2 \left(2f^2g^2 + k^2K \right) + \left(Q_3Q_4 - 4e^{-hf}e^{-hg} \right) fg \left(g^2 + 3k^2 \right) \right\} \quad (25)$$

$$S_{44} = -gK_1(Q_2Q_3fg - Q_1Q_4k^2) \quad (26)$$

where,

$$K_1 = k^2 - g^2 \quad (27)$$

$$D = Q_1 Q_2 (f^2 g^2 + k^4) - 2(Q_3 Q_4 - 4e^{-hf} e^{-hg}) f g k^2 \quad (28)$$

$$Q_1 = 1 - e^{-2hf} \quad (29)$$

$$Q_2 = 1 - e^{-2hg} \quad (30)$$

$$Q_3 = 1 + e^{-2hf} \quad (31)$$

and,

$$Q_4 = 1 + e^{-2hg} \quad (32)$$

Similarly, the closed form stiffness matrix of a 1-noded element (i.e., a layer with an infinite thickness, thus only having the upper boundary), **S_{1-noded}**, is obtained as the following,

$$\mathbf{S}_{\text{1-noded}} = \frac{s\hat{\mu}}{D^1} \cdot \begin{bmatrix} S_{11}^1 & S_{12}^1 \\ Sym & S_{22}^1 \end{bmatrix} \quad (33)$$

whose elements, S_{11}^1 through S_{22}^1 are given as the following.

$$S_{11}^1 = 2k^2 f - fK \quad (34)$$

$$S_{12}^1 = -2kfg + kK \quad (35)$$

$$S_{22}^1 = 2k^2 g - gK \quad (36)$$

where,

$$D^1 = k^2 - fg \quad (37)$$

Global Stiffness Matrix

After the stiffness matrices have been obtained for all the layers that make up the structure, the global stiffness matrix must be constructed in the same way as the traditional FEM methods

(Cook et. al., 2001). Figure 10 shows the schematics of the global stiffness matrices for the two types of layered structures that are most widely adopted for modeling a pavement system.

Figure 10(a) shows how the global stiffness matrix is constructed for a layered system resting on a halfspace (i.e., 1-noded semi-infinite element). On the other hand, Figure 10(b) shows the global stiffness matrix that can be used for a layered system sitting on a stiff bedrock at shallow depth.

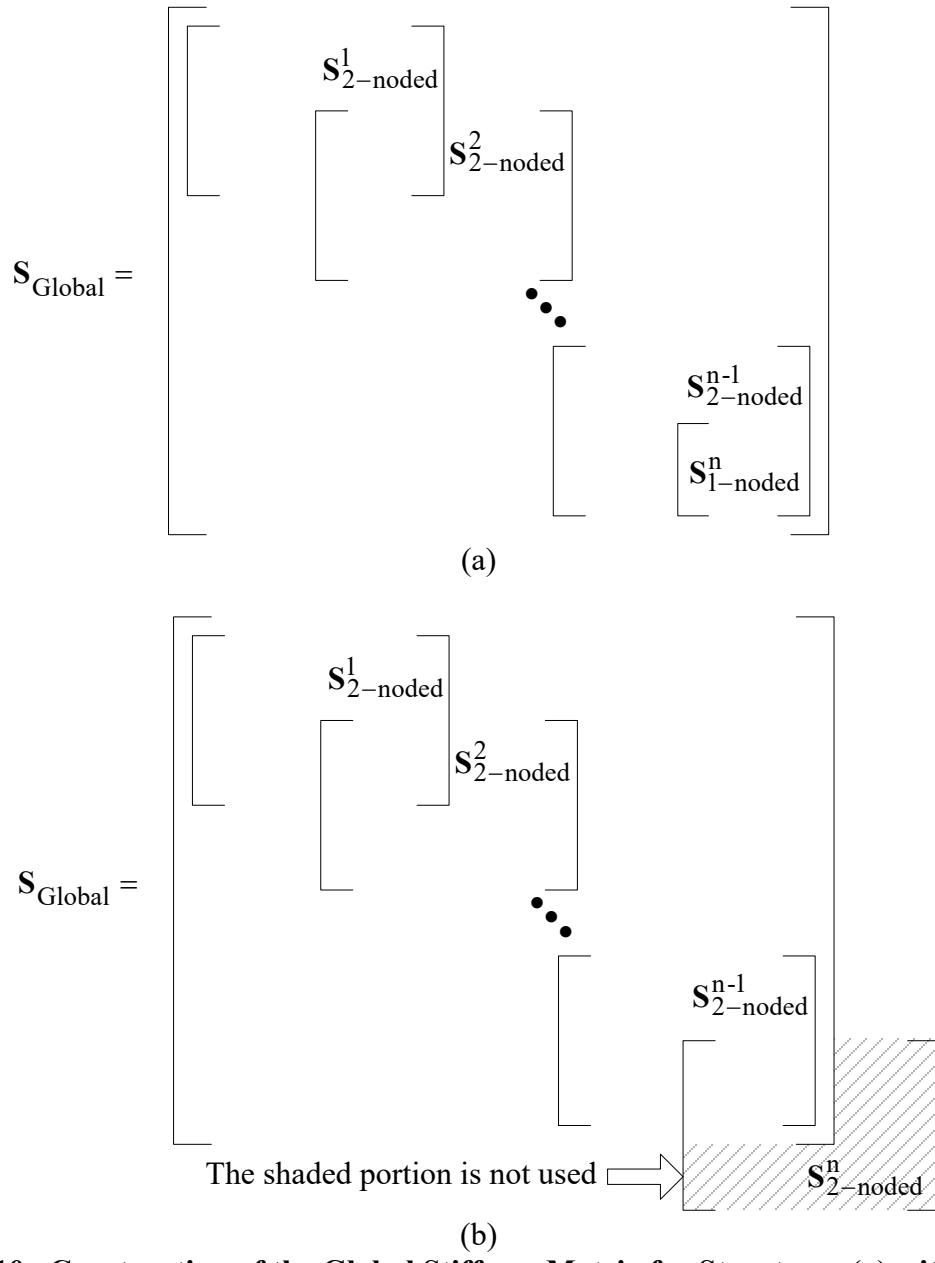


Figure 10. Construction of the Global Stiffness Matrix for Structures (a) with and (b) without a halfspace

Upon constructing the global stiffness matrix, the displacements at the system nodes can be found from the following equation:

$$\bar{\mathbf{U}} = \mathbf{S}_{\text{Global}}^{-1} \cdot \bar{\mathbf{P}} \quad (38)$$

where,

$$\bar{\mathbf{U}} = \{\bar{U}_{r1} \quad \bar{U}_{z1} \quad \dots \quad \bar{U}_{ri} \quad \bar{U}_{zi} \quad \dots \quad \bar{U}_{rn} \quad \bar{U}_{zn}\}' \quad (39)$$

is a vector of system displacements to be calculated in global coordinates with \bar{U}_{ri} and \bar{U}_{zi} being the radial and vertical displacements at the i^{th} node from the top, respectively. Similarly,

$$\bar{\mathbf{P}} = \{\bar{P}_{r1} \quad \bar{P}_{z1} \quad \dots \quad \bar{P}_{ri} \quad \bar{P}_{zi} \quad \dots \quad \bar{P}_{rn} \quad \bar{P}_{zn}\}' \quad (40)$$

is a nodal force vector in global coordinates, with the radial and vertical forces at the i^{th} node denoted as \bar{P}_{ri} and \bar{P}_{zi} , respectively. The nodal forces in this vector should be obtained from the boundary conditions as will be presented in the next section.

Boundary Conditions for a Circular Unit Impulse Loading at the Ground Surface

For the loading that is induced by an impact of a falling weight at the ground surface, all components of $\bar{\mathbf{P}}$ in Equation (40) vanish except for \bar{P}_{z1} . In other words, the only external load applied to the system is in the vertical direction at the top node (node 1). For the formulation of ViscoWave, the boundary condition will be a surface force in the form of a unit impulse load acting over a circular area. In the physical time and spatial domain, this boundary condition is mathematically expressed as the following:

$$P_{z1}(r, t) = R(r) \cdot \delta(t) \quad (41)$$

where $\delta(t)$ is the dirac delta function for the impulse loading and,

$$R(r) = \begin{cases} 1 & , 0 < r \leq a \\ 0 & , r > a \end{cases} \quad (42)$$

where a is the radius of the circular loaded area. Since the stiffness matrices were derived in the Laplace-Hankel domain rather than the physical domain, it is also necessary to convert the above boundary condition into the one in the transformed domain. Because the Laplace transform of $\delta(t)$ is equal to 1, taking the Laplace-Hankel transforms on Equation (5-61) simply results in the following equation:

$$\bar{P}_{z1}(k, s) = \frac{a}{k} J_1(ka) \quad (43)$$

where J_1 is the Bessel function of order one.

Numerical Inversion of Laplace and Hankel Transforms

The displacements at all nodes of the system obtained from Equation (38) are in the Laplace-Hankel domain and need to be inverse transformed back to the physical domain. However, due to the mathematical complexity arising from the viscoelastic material behavior and the wave propagation phenomenon, closed form equations cannot be obtained for these inverse transforms. As such, the inversions are carried out numerically for both the Laplace and Hankel transforms.

The closed form equation for the inverse Hankel transform of the vertical displacement at node i is given as:

$$\hat{U}_{zi}(r) = \int_0^{\infty} \bar{U}_{zi}(k) J_0(kr) k dk \quad (44)$$

where J_0 is the Bessel function of order zero. The above integral can also be written as a series of integrals:

$$\hat{U}_{zi}(r) = \int_{b_1}^{b_2} \bar{U}_{zi}(k) J_0(kr) k dk + \int_{b_2}^{b_3} \bar{U}_{zi}(k) J_0(kr) k dk + \dots + \int_{b_n}^{b_{n+1}} \bar{U}_{zi}(k) J_0(kr) k dk + \dots \quad (45)$$

In reference to Cornille (1972), the limits of each integration, b_n , are selected to be at the first 5 successive roots of the Bessel function of order one (J_1). This also means that Equation (45) is evaluated only for the first five cycles of the Bessel function. The roots of J_1 (hard-coded into ViscoWave) are provided in the following table.

Table 2. First Five Roots of J_1 (b_n Values Hard-Coded into ViscoWave)

J_1 Root Number	Root Value	J_1 Root Number	Root Value
0	0.0	3	10.1734681350627
1	3.83170597020751	4	13.3236919363142
2	7.01558666981562	5	16.4706300508776

To improve accuracy of the numerical inversion in ViscoWave, the region between the successive roots of J_1 (e.g., from b_1 to b_2) are subdivided into smaller regions (up to ten smaller regions) of equal intervals.

Each integral in the right-hand side of Equation (45) needs to be evaluated numerically. Upon selecting the 6-point Gaussian quadrature as the numerical scheme to be used, the integral in the above equation can be evaluated as (Abramowitz and Stegun, 1972):

$$\int_{b_n}^{b_{n+1}} \bar{U}_{zi}(k) J_0(kr) k dk = \frac{b_{n+1} - b_n}{2} \sum_{p=1}^6 w_p \bar{U}_{zi}(\beta_p) J_0(\beta_p) \beta_p \quad (46)$$

where

$$\beta_p = \left(\frac{b_{n+1} - b_n}{2} \right) x_p + \left(\frac{b_{n+1} + b_n}{2} \right) \quad (47)$$

and x_p and w_p are the Gaussian nodes and their corresponding weights, respectively. Table 3 shows the values (or coordinates) and weights of the Gaussian nodes that are hard-coded into ViscoWave for the purpose of 6-point Gaussian quadrature.

Table 3. Coordinates and Weights of 6-Point Gaussian Quadrature Nodes

Gaussian Node Number	Node Coordinate	Node Weight
1	-0.932469514203152	0.17132449237917
2	-0.661209386466265	0.360761573048139
3	-0.238619186083197	0.467913934572691
4	0.238619186083197	0.467913934572691
5	0.661209386466265	0.360761573048139
6	0.932469514203152	0.17132449237917

 The above roots of Bessel function J_1 as well as the Gaussian nodes and weights are stored in the c++ header file named ViscoWaveEngine.h.

For the inverse Laplace transform, a multi precision numerical scheme known as the Fixed Talbot Algorithm (Abate and Valko, 2004) is implemented into ViscoWave. Fixed Talbot Algorithm is a numerical method for approximating the Bromwich integral (which is the theoretical equation for the inverse Laplace transform) using the trapezoidal rule. The inversion is computed using the following equation.

$$U_{zi}(t) = \frac{\alpha}{M} \left\{ \frac{1}{2} \hat{U}_{zi}(\alpha) e^{\alpha t} + \sum_{q=1}^{M-1} \operatorname{Re} \left[e^{ts(\theta_q)} \hat{U}_{zi}(s(\theta_q)) (1 + j\gamma(\theta_q)) \right] \right\} \quad (48)$$

where

$$\alpha = \frac{2M}{5t} \quad (49)$$

$$\theta_q = \frac{q\pi}{M} \quad (50)$$

$$s(\theta) = \alpha \theta (\cot \theta + j), \quad -\pi < \theta < +\pi \quad (51)$$

$$\gamma(\theta) = \theta + (\cot \theta - 1) \cot \theta \quad (52)$$

and $j = \sqrt{-1}$. In the above equations, M is the number of precision decimal digits to be used for the numerical analysis.

 In ViscoWave, the precision decimal digits, M , is hard-coded to a value of 16.

System Response To Arbitrary Loading

As described previously, the boundary condition considered in Equation (41) and Equation (43) was for a **unit impulse load** distributed over a circular area. As such, the vertical displacement, U_{zi} , obtained in Equation (48) represents the **unit impulse response** of the layered system in time domain. The primary advantage of the time domain unit impulse response is that the system response to any arbitrary loading can be obtained through the convolution integral (Santamarina and Fratta, 1998, Bendat and Piersol, 2010). Theoretically, this convolution integral for a continuous function is given as:

$$y_{zi}(t) = U_{zi}(t) * F(t) = \int_0^t U_{zi}(t - \tau) F(\tau) d\tau \quad (53)$$

where $F(t)$ may be any arbitrary time dependent loading function and $y_{zi}(t)$ is the corresponding vertical displacement at node i . For a discrete signal such a FWD time history, the above equation needs to be evaluated numerically as (Santamarina and Fratta, 1998, Bendat and Piersol, 2010):

$$y_{zi}(t_n) = \sum_{t_p=1}^{t_n} U_{zi}(t_n - t_p) F(t_p) \Delta t \quad (54)$$

where Δt is time interval of the discrete signal and $t_n = n\Delta t$ for an integer n .

INTERPOLATION OF IMPULSE RESPONSE USING CUBIC SPLINE

The numerical inversion of Laplace and Hankel transforms described above requires a significant amount of looping. For instance, consider the following typical case.

- 300 discrete time history data points: FWD time history from 0.0 ms to 59.8 ms at an interval of 0.2 ms.
- Up to 50 intervals for Hankel transform inversion: For the first five roots of J_1 with the interval subdivided into 10 smaller intervals.
- 6-point Gaussian quadrature for numerical Hankel transform inversion.
- Precision decimal digits, M , set to a value of 16

Given the above scenario, it is necessary that the global stiffness matrix (Equation (38)) needs to be constructed and inverted 1.44 million times ($= 300 \times 50 \times 6 \times 16$) to obtain the simulation results for a single FWD time history (even without counting the number of geophone sensors).

To improve the efficiency of ViscoWave runtime, an interpolation scheme known as the Cubic Spline (CS) has been implemented. Essentially, this method is based on third-order polynomials (i.e., splines) that are constructed in a piecewise manner to pass through a set of control (i.e., known) points. The function values in between the control points are then approximated from the piecewise splines.

In ViscoWave, the impulse response is evaluated at a total of 29 control points (along the time scale) for constructing the splines. These control time points (hard-coded into ViscoWave) are shown in Table 4.

Table 4. Control Time Points in ViscoWave for Evaluating Impulse Response

Control Point Number	Time Value (ms)	Control Point Number	Time Value (ms)
1	10^{-6}	16	10.0
2	0.2	17	12.0
3	0.4	18	14.0
4	0.6	19	16.0
5	0.8	20	18.0
6	1.0	21	20.0
7	1.5	22	25.0
8	2.0	23	30.0
9	2.5	24	35.0
10	3.0	25	40.0
11	4.0	26	50.0
12	5.0	27	60.0
13	6.0	28	69.8
14	7.0	29	70.0
15	8.0		

 The Cubic Spline (CS) Interpolation was not implemented in the original version of ViscoWave by Lee (2013, 2014), and is a relatively recent update that has never been documented previously. The use of CS method allows for calculating the true impulse response at 29 control points (rather than 300 discrete time points or more). This resulted in a significant improvement in time-efficiency of ViscoWave (approximately 10 times faster).

After the impulse response has been evaluated at the above control points, the impulse response at any given time (between 0.0 ms and 70.0 ms) is interpolated using the CS method. As an example, Figure 11 shows the impulse response (at the center of the load plate, D0, and at an offset of 60 in. from the load plate, D60) generated at the control points and interpolated using the CS method. Note that the splines are generated such that not only the function value (in this

case the impulse response or deflection) but also the first derivative of the function (in this case the slope of the impulse response) is continuous before and after a control point, which allows for producing a smooth, interpolated impulse response.

Figure 12 shows the comparison between the impulse response obtained from CS interpolation to those obtained without interpolation (i.e., the true impulse response), which shows excellent agreement between the two responses.

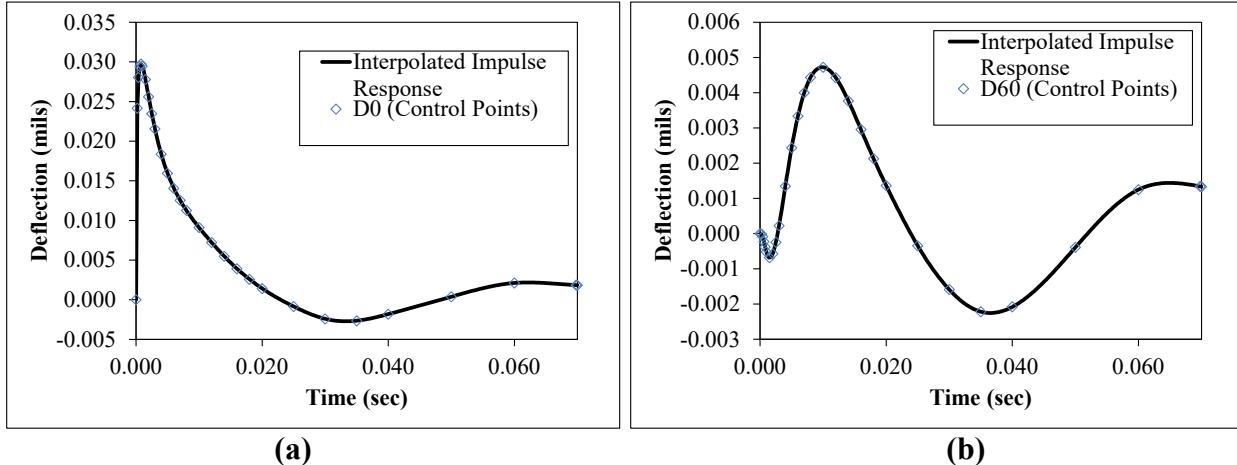


Figure 11. Impulse Response at Control Points and Interpolated Impulse Response for (a) D0 and (b) D60

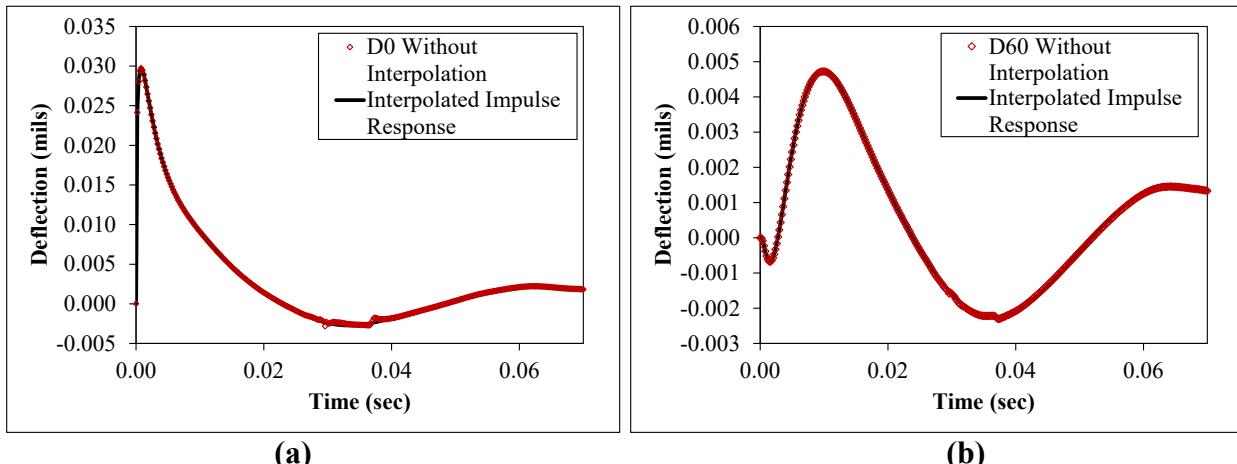


Figure 12. Impulse Responses with and without CS Interpolation for (a) D0 and (b) D60

MATERIAL MODELS (ELASTIC AND VISCOELASTIC)

Two material models are available and implemented into ViscoWave, namely linear elastic and linear viscoelastic models. Since the global stiffness matrix is constructed in the Laplace-Hankel transformed domain, it is necessary that the material properties (such as modulus) be transformed appropriately.

For a homogenous, isotropic, **elastic material** whose properties are independent of time, the relationship between the elastic modulus, E , and the shear modulus, μ , is given by the theory of linear elasticity as:

$$\mu = \frac{E}{2(1 + \nu)} \quad (55)$$

Because the parameters in the above equation are not functions of time, the Laplace transform of the above equation is simply obtained as:

$$\hat{\mu}(s) = \frac{\mu}{s} = \frac{E}{2(1 + \nu)} \cdot \frac{1}{s} \quad (56)$$

However, as it was noted by the pioneer of the spectral element method for layered media (Rizzi, 1989), it is advantageous to add a small amount of damping to the lamé constant, μ , as no realistic material is purely elastic. Following Rizzi (1989), this artificial damping can be added to the above lamé constant as:

$$\hat{\mu}(s) = \hat{\mu}(s) \cdot (1 + \zeta \cdot s) \quad (57)$$

where ζ is a damping constant. To simulate the wave propagation through an elastic layer, the above simple equation can be substituted into the equations for the layer elements presented earlier.

 For linear elastic materials, it is recommended that a small amount of damping be used. The default value for the damping constant in ViscoWave is 0.1 percent (i.e., 0.001).

The modulus of a **viscoelastic material** is time (or equivalently, frequency) dependent. A well-known model for the time (or frequency) dependency of a viscoelastic modulus is given in terms of a sigmoidal function of the following form, which is implemented into ViscoWave.

$$\log E(t) = c_1 + \frac{c_2}{1 + e^{c_3 + c_4 \log(t)}} \quad (58)$$

where $E(t)$ is the time dependent, viscoelastic relaxation modulus at time t , and c_1 through c_4 are model constants.

 In the original version of ViscoWave by Lee (2013, 2014), creep compliance was used to for the viscoelastic material model (rather than relaxation modulus). However, due to the sigmoidal model for the relaxation modulus gaining more popularity (e.g., in Mechanistic-Empirical Pavement Design), the creep compliance model has been replaced by the sigmoidal relaxation modulus in the current version of ViscoWave.

One drawback of the sigmoidal function shown in Equation (58) is that its Laplace transform does not exist in closed form. As such, ViscoWave internally utilizes the Prony series model for the relaxation modulus. The time dependent Prony series model for the relaxation modulus is given as the following.

$$E(t) = E_0 + \sum_{i=1}^M E_i \cdot e^{-\frac{t}{\rho_i}} \quad (59)$$

where E_0 and E_i are the Prony coefficients to be determined by fitting the above function on top the sigmoidal model, M is the number of Prony elements, and ρ_i is the relaxation time for the i^{th} Prony element.

For fitting the above Prony function to the sigmoidal function in Equation (58), the number of Prony elements (M) and the relaxation times (ρ_i) are usually specified, such that the only unknowns to be determined are the Prony coefficients (E_i). In ViscoWave, a total of 14 Prony elements are used (i.e., $M = 14$) with the following relaxation times.

$$\rho_i = 10^{(8-i)}, \quad i = 1, 2, \dots, M \quad (60)$$

With the above relaxation times specified (i.e., built into ViscoWave), the Prony coefficients are obtained using the method of Linear Least Squares (LLS). In order to use the LLS, it is necessary that a fairly large number of known $E(t)$ values (i.e., the values that are to be fitted over Equation (59)) are made available over a wide range of time values. The time values at which the relaxation modulus values are to be fitted, are given as the following.

$$t_k = 10^{a+bk}, \quad k = 0, 1, 2, \dots, n \quad (61)$$

where $a = -6.7$, $b = 0.02$, and n is the total number of discrete time points (equal to 670 in ViscoWave). For each time value, t_k , given in Equation (61), the corresponding relaxation modulus, E_k , can be obtained simply from Equation (58), or more specifically, from the following.

$$E_k = E(t_k) = 10^{\frac{c_1 + \frac{c_2}{1 + e^{c_3 + c_4 \log(t_k)}}}{}} \quad (62)$$

The objective of the LLS is to find the Prony coefficients (E_i) that produce the minimum error (or difference) between the E_k values computed from the sigmoidal function (Equation (62)) and those calculated using Prony series (Equation (59)). Mathematically, this means the goal of LLS is to minimize the value of the following objective function, F .

$$F = \sum_{k=0}^n \left(E_k - E_0 - \sum_{i=1}^M E_i \cdot e^{-\frac{t_k}{\rho_i}} \right)^2 \quad (63)$$

Although it may look complicated, the above equation is nothing more than a quadratic function of real-valued variables. In other words, the minimum of the objective function, F , can be found when its partial derivative with respect to each of the unknowns (i.e., E_i 's) are set to zero. Taking the partial derivative of F with respect to E_0 can be obtained as the following:

$$\frac{\partial F}{\partial E_0} = -2 \sum_{k=0}^n \left(E_k - E_0 - \sum_{i=1}^M E_i \cdot e^{-\frac{t_k}{\rho_i}} \right) = 0 \quad (64)$$

which can be reorganized such that all the unknowns are on the left-hand side of the equation. That is:

$$nE_0 + E_1 \sum_{k=0}^n e^{-\frac{t_k}{\rho_1}} + E_2 \sum_{k=0}^n e^{-\frac{t_k}{\rho_2}} + \dots + E_M \sum_{k=0}^n e^{-\frac{t_k}{\rho_M}} = \sum_{k=0}^n E_k \quad (65)$$

Similarly, the general form for the partial derivative of F with respect to any E_i is obtained as the following.

$$\frac{\partial F}{\partial E_i} = -2e^{-\frac{t_k}{\rho_i}} \cdot \sum_{k=0}^n \left(E_k - E_0 - \sum_{i=1}^M E_i \cdot e^{-\frac{t_k}{\rho_i}} \right) = 0 \quad (66)$$

Note that the above equation represents a total of M equations for i ranging from 1 to M . As an example, if $i = 1$, the above equation is written as the following, after reorganization.

$$E_0 \sum_{k=0}^n e^{-\frac{t_k}{\rho_1}} + E_1 \sum_{k=0}^n e^{-\frac{t_k}{\rho_1}} \cdot e^{-\frac{t_k}{\rho_1}} + \dots + E_M \sum_{k=0}^n e^{-\frac{t_k}{\rho_1}} \cdot e^{-\frac{t_k}{\rho_M}} = \sum_{k=0}^n E_k e^{-\frac{t_k}{\rho_1}} \quad (67)$$

As another example, if $i = M$, Equation (66) can be written as the following.

$$E_0 \sum_{k=0}^n e^{-\frac{t_k}{\rho_M}} + E_1 \sum_{k=0}^n e^{-\frac{t_k}{\rho_M}} \cdot e^{-\frac{t_k}{\rho_1}} + \dots + E_M \sum_{k=0}^n e^{-\frac{t_k}{\rho_M}} \cdot e^{-\frac{t_k}{\rho_M}} = \sum_{k=0}^n E_k e^{-\frac{t_k}{\rho_M}} \quad (68)$$

Collectively, Equation (65) and the M equations represented in Equation (66) can be written as the following in matrix form.

$$\begin{bmatrix} n & \sum_k e^{-\frac{t_k}{\rho_1}} & \dots & \sum_k e^{-\frac{t_k}{\rho_M}} \\ \sum_k e^{-\frac{t_k}{\rho_1}} & \sum_k e^{-\frac{t_k}{\rho_1}} \cdot e^{-\frac{t_k}{\rho_1}} & \sum_k e^{-\frac{t_k}{\rho_1}} \cdot e^{-\frac{t_k}{\rho_M}} & \vdots \\ \vdots & \ddots & \vdots & \\ \sum_k e^{-\frac{t_k}{\rho_M}} & \sum_k e^{-\frac{t_k}{\rho_M}} \cdot e^{-\frac{t_k}{\rho_1}} & \dots & \sum_k e^{-\frac{t_k}{\rho_M}} \cdot e^{-\frac{t_k}{\rho_1}} \end{bmatrix} \begin{Bmatrix} E_0 \\ E_1 \\ \vdots \\ E_M \end{Bmatrix} = \begin{Bmatrix} \sum_k E_k \\ \sum_k E_k e^{-\frac{t_k}{\rho_1}} \\ \vdots \\ \sum_k E_k e^{-\frac{t_k}{\rho_M}} \end{Bmatrix} \quad (69)$$

Denoting the M -by- M matrix on the left-hand side of Equation (69) as $\mathbf{A}_{M \times M}$ and the vector on the right hand side as $\mathbf{B}_{M \times 1}$, the above equation can be written simply as the following.

$$\mathbf{A}_{M \times M} \cdot \{E_0 \quad E_1 \quad \dots \quad E_M\}^T = \mathbf{B}_{M \times 1} \quad (70)$$

Since there are no unknowns included in any of the elements for $\mathbf{A}_{M \times M}$ and $\mathbf{B}_{M \times 1}$, they can easily be computed using a computer algorithm or software. Once $\mathbf{A}_{M \times M}$ and $\mathbf{B}_{M \times 1}$ are constructed, the vector of unknowns (i.e., the Prony coefficients) can be obtained by inverting the M -by- M matrix.

$$\{E_0 \quad E_1 \quad \dots \quad E_M\}^T = \mathbf{A}_{M \times M}^{-1} \cdot \mathbf{B}_{M \times 1} \quad (71)$$

 In many mathematical programming languages, the method of Linear Least Squares (LLS) is readily available for immediate implementation. In other words, the matrix $\mathbf{A}_{M \times M}$ and the vector $\mathbf{B}_{M \times 1}$ are (automatically) constructed within the LLS routine and inverted to produce the vector of unknowns, provided that appropriate inputs are provided for the LLS routine.

The LLS routine used by ViscoWave is the “lsfitlinear” function provided by the open-source ALGLIB C++ language package.

 **The primary advantage of LLS is that it does not require any knowledge (or initial guess) for the unknowns that are typically needed for Nonlinear Least Squares (NLS),** and that it is generally faster than NLS as it only requires a single, one-time inversion of the M -by- M matrix.

Once the Prony coefficients are obtained from Equation (71), these coefficients can easily be substituted back into Equation (59), which completes the Prony fitting. As an example, Figure 13 shows an example of the relaxation modulus plots from both the sigmoidal function as well as the Prony series, which shows an excellent agreement between them.

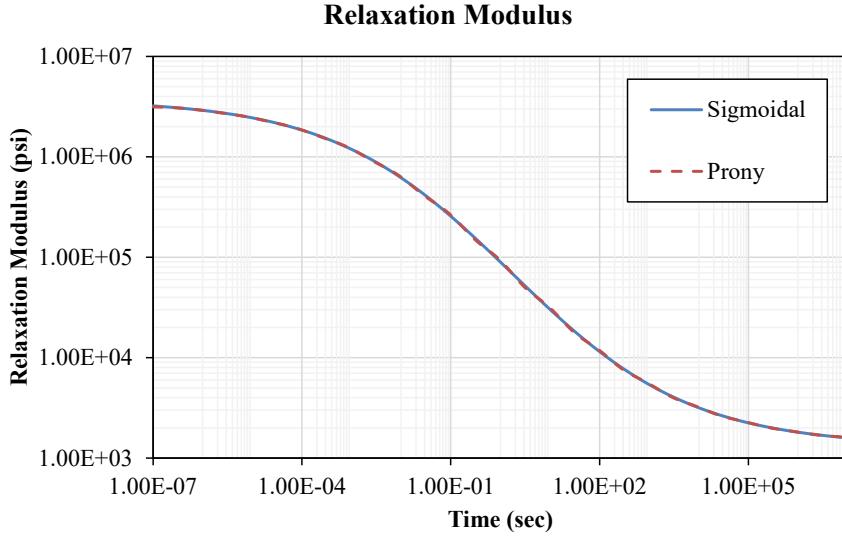


Figure 13. Impulse Responses with and without CS Interpolation for (a) D0 and (b) D60

Recall that the purpose of representing the relaxation modulus in Prony series was because the Laplace transform of the sigmoidal function in Equation (58) cannot be derived in closed form. The Laplace transform of Prony series shown in Equation (59) is straight forward, and can be obtained as the following.

$$\begin{aligned}\hat{E}(s) &= \frac{E_0}{s} + \frac{\rho_1 E_1}{\rho_1 s + 1} + \frac{\rho_2 E_2}{\rho_2 s + 1} + \dots + \frac{\rho_M E_M}{\rho_M s + 1} \\ &= \frac{E_0}{s} + \sum_{i=1}^M \frac{\rho_i E_i}{\rho_i s + 1}\end{aligned}\quad (72)$$

Then, with the assumption that the Poisson's ratio of the viscoelastic material is independent of time (which usually is a good assumption), the Laplace transformed, viscoelastic shear modulus can be obtained conveniently as the following.

$$\hat{\mu}(s) = \frac{\hat{E}(s)}{2(1+\nu)} = \frac{1}{2(1+\nu)} \left\{ \frac{E_0}{s} + \sum_{i=1}^M \frac{\rho_i E_i}{\rho_i s + 1} \right\} \quad (73)$$

The above shear modulus is used for constructing the element stiffness matrix of a viscoelastic layer in Equation (16).

INTERCONVERSION BETWEEN RELAXATION MODULUS AND DYNAMIC MODULUS

Thus far, the viscoelastic material effects were addressed in terms of the relaxation properties (i.e., uniaxial relaxation modulus, $E(t)$, or shear relaxation modulus, $\mu(t)$) as functions of time. This is primarily because of the close relationship between the relaxation modulus in time domain (i.e., Equation (59)) and its counterpart in the Laplace domain (i.e., Equation (72)).

In the field of asphalt materials and pavement engineering, the viscoelastic behaviour of AC material is also frequently described in terms of **Dynamic Modulus** which is dependent on frequency (rather than time). But, it is first necessary to discuss the **Complex Modulus** of a viscoelastic material defined as the following.

$$E^*(\omega) = E'(\omega) + E''(\omega) = \frac{\sigma^*}{\varepsilon_0} = \frac{\sigma_0}{\varepsilon_0} \cdot e^{i\delta} \quad (74)$$

where $E'(\omega)$ and $E''(\omega)$ are the real and imaginary parts of the complex modulus that depend on the angular frequency ω , σ^* and σ_0 are the complex- and real-valued stress amplitudes, respectively, ε_0 is the real-valued strain amplitude, and δ is the phase angle. Given the above complex modulus, the **Dynamic Modulus** is defined as the magnitude of the complex modulus given as the following equation (Kim, 2009).

$$|E^*(\omega)| = \sqrt{\{E'(\omega)\}^2 + \{E''(\omega)\}^2} \quad (75)$$

Similar to the time dependent relaxation modulus, a sigmoidal function is frequently used in the pavement engineering community for modelling the frequency dependent dynamic modulus. The sigmoidal function essentially has the same form as Equation (58), but with a new set of coefficients. That is:

$$\log(|E^*|) = d_1 + \frac{d_2}{1 + e^{d_3 + d_4 \log(f)}} \quad (76)$$

In order to explain the procedure for the interconversion of the viscoelastic modulus (i.e., converting from relaxation modulus in Equation (58) to dynamic modulus in Equation (76), or vice versa), it is necessary to understand the effect of the sigmoidal coefficients (i.e., c_1 through c_4 , or d_1 through d_4). In reference to Figure 14, the following describes how the sigmoidal function for the relaxation modulus (in Log-Log scale) is affected by the respective coefficients.

1. The coefficient c_1 defines the **Lower Asymptote** of the curve.
2. The coefficient c_2 defines the **difference between the Upper and the Lower Asymptotes** of the curve. I.e., the Upper Asymptote value is equal to $c_1 + c_2$.
3. The coefficient c_3 shifts the entire curve **horizontally** (left or right depending on its value).

4. The coefficient c_4 defines the **slope of the curve** (i.e., how steep or how fast it increases) between the lower and the upper asymptotes.

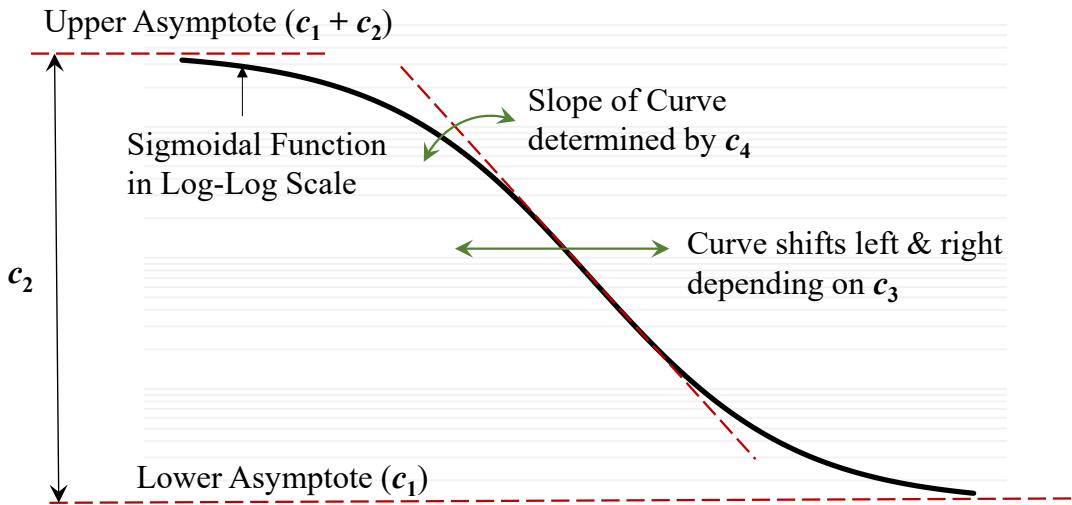


Figure 14. Effect of Sigmoidal Coefficients on Relaxation Modulus Curve

The above illustration was limited to the relaxation modulus and the coefficients c_1 through c_4 . In the case of Dynamic Modulus, the curve will appear as a mirrored version (i.e., flipped left-to-right) of the curve shown above. **The parameters d_1 through d_4 , thus have similar effects on the curve as their counterparts, c_1 through c_4 .**

Many procedures and approximation methods have been proposed for interconversion of the viscoelastic material properties from time domain to frequency domain, and vice versa. Two of the most frequently used approximations are derived from the direct relationship between time and frequency values.

As an example, some researchers have used the inverse relationship between time and ordinary frequency (i.e., $t = 1/f$) for the interconversion. Substituting this relationship into Equation (58) for an approximate interconversion results in the following.

$$\begin{aligned} \log(|E^*|)_{\text{Approx}} &= c_1 + \frac{c_2}{1 + e^{c_3 + c_4 \log(1/f)}} \\ &= c_1 + \frac{c_2}{1 + e^{c_3 - c_4 \log(f)}} \end{aligned} \quad (77)$$

Comparing the above equation with Equation (76) reveals the following coefficient relationships for the interconversion.

$$d_1 = c_1 \quad (78-1)$$

$$d_2 = c_2 \quad (78-2)$$

$$d_3 = c_3 \quad (78-3)$$

$$d_4 = -c_4 \quad (78-4)$$

Said differently, if the relationship $t = 1/f$ is used, the interconversion is approximated simply by changing the sign of the coefficient c_4 (if going from relaxation to dynamic modulus) or d_4 (if going from dynamic modulus to relaxation modulus).

Another commonly used time-frequency relationship is from the definition of the angular frequency (i.e., $t = 1/2\pi f$). Substituting this relationship into Equation (58) results in the following.

$$\begin{aligned} \log(|E^*|)_{Approx} &= c_1 + \frac{c_2}{1 + e^{c_3 + c_4 \log(1/2\pi f)}} \\ &= c_1 + \frac{c_2}{1 + e^{c_3 - c_4 \log(2\pi) - c_4 \log(f)}} \end{aligned} \quad (79)$$

Comparing the above equation with Equation (76) reveals the following coefficient relationships for the interconversion.

$$d_1 = c_1 \quad (80-1)$$

$$d_2 = c_2 \quad (80-2)$$

$$d_3 = c_3 - c_4 \log(2\pi) \quad (80-3)$$

$$d_4 = -c_4 \quad (80-4)$$

The above indicates that if the relationship $t = 1/2\pi f$ is used, the interconversion is approximated by changing the sign of the coefficient c_4 , then by shifting the entire curve horizontally by the amount $-c_4 \log(2\pi)$.

NEITHER of the above relationship is used in ViscoWave Interface (or more specifically, the “Dynamic_Modulus_Calc” tab of ViscoWave spreadsheet). Instead, ViscoWave utilizes the Prony series expressions for interconversion. This is because the complex modulus (and hence the dynamic modulus) is well defined according to the theory of viscoelasticity. More specifically, the Prony series representation of real and imaginary parts of the complex modulus are given as the following.

$$E'(\omega) = E_0 + \sum_{i=1}^M \frac{\varpi^2 \rho_i^2 E_i}{\varpi^2 \rho_i^2 + 1} \quad (81-1)$$

$$E''(\omega) = \sum_{i=1}^M \frac{\varpi \rho_i E_i}{\varpi^2 \rho_i^2 + 1} \quad (81-2)$$

Note that the Prony coefficients in the above equations (i.e., E_0 and E_i values) are the same coefficients used in the Prony representation of the relaxation modulus (i.e., Equation (59)). In other words, if the Prony coefficients were determined from the relaxation modulus, the dynamic modulus of the same material can be determined as the following for any given angular frequency, ω (by means of Equation (75) and Equation (81)).

$$|E^*(\omega)| = \sqrt{\left\{E_0 + \sum_{i=1}^M \frac{\varpi^2 \rho_i^2 E_i}{\varpi^2 \rho_i^2 + 1}\right\}^2 + \left\{\sum_{i=1}^M \frac{\varpi \rho_i E_i}{\varpi^2 \rho_i^2 + 1}\right\}^2} \quad (82)$$

Since the Prony series representation of the relaxation modulus (Equation (59)) and the dynamic modulus (Equation (82)) are well established in the theory of viscoelasticity, ViscoWave utilizes these equations as the “Ground Truth” for interconversion of the viscoelastic properties represented in terms of the sigmoidal functions. Furthermore, the following conditions, that will simplify the interconversion of sigmoidal functions, are applied to the sigmoidal coefficients.

1. For the same material, the lower and the upper asymptotes should not change whether the modulus is expressed in time-domain or in frequency-domain. Therefore, in reference to Figure 14, the following must be true for a given viscoelastic material.

$$d_1 = c_1 \quad (83-1)$$

$$d_2 = c_2 \quad (83-2)$$

Note that these conditions are also observed in the approximate conversions, Equations (78-1), (78-2), (80-1), and (80-2).

2. The exact relation between time and frequency is not defined specifically, but in a generic manner that specifies the inverse relation between the two variables. Thus:

$$t = \frac{1}{C \cdot f} \quad (84)$$

where C is an arbitrary constant. Note that if $C = 1$, then the above equation reduces to the time-frequency relationship using the ordinary frequency (i.e., $t = 1/f$) and if $C = 2\pi$, then the above equation becomes the time-frequency relationship using the angular frequency ($t = 1/2\pi f$). Substituting this relationship into Equation (58) results in the following.

$$\begin{aligned} \log(|E^*|) &= c_1 + \frac{c_2}{1 + e^{c_3 + c_4 \log(1/Cf)}} \\ &= c_1 + \frac{c_2}{1 + e^{c_3 - c_4 \log(C) - c_4 \log(f)}} \end{aligned} \quad (85)$$

Comparing the above equation with Equation (76) reveals the following coefficient relationships for the interconversion.

$$d_3 = c_3 - c_4 \log(C) \quad (86-1)$$

$$d_4 = -c_4 \quad (86-2)$$

Equation (83) and Equation (84) indicate that if the sigmoidal coefficients for relaxation modulus are known, three out of four sigmoidal coefficients for dynamic modulus are determined easily. Then, the only unknown is the d_3 coefficient (or c_3 coefficient if converting from dynamic modulus to relaxation modulus) due to the arbitrary constant C . For the interconversion, the constant C is not determined, and the d_3 coefficient (or c_3 coefficient) is determined directly by fitting the sigmoidal function to Prony series. These procedures are explained in the following.

Conversion from Relaxation Modulus (Time Dependent) to Dynamic Modulus (Frequency Dependent)

The following procedure is used in ViscoWave for converting the sigmoidal function for relaxation modulus to the sigmoidal function for dynamic modulus.

1. Since the sigmoidal coefficients for the relaxation modulus (c_1 through c_4 of Equation (58)) are known, determine the Prony coefficients by fitting Equation (59) to Equation (58) by means of LLS described in the previous section.
2. Substitute the Prony coefficients into Equation (82) to construct the dynamic modulus from Prony series.
3. Construct the dynamic modulus using Equation (76), with the values of d_1 , d_2 , and d_4 determined from Equations (83-1), (83-2), and (86-2), respectively. **For the unknown d_3 value, assume it is equal to c_3 (for now).** Figure 15 shows how this sigmoidal function compares to the Prony series dynamic modulus.
4. Using the above initial guess for d_3 , run NLS to determine the d_3 value that minimizes the error between the dynamic modulus curves from Prony series and sigmoidal function. Figure 16 shows a comparison between the post-NLS sigmoidal function and the Prony series dynamic modulus.

 For the NLS routine, ViscoWave uses the SOLVER application that is built into MS Excel.

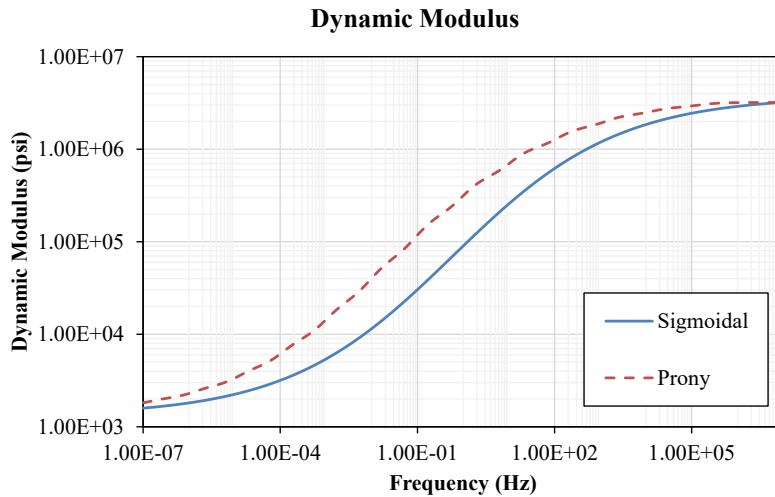


Figure 15. Dynamic Modulus Curves from Prony Series and Sigmoidal Function Using the Initial Guess of $d_1 = c_1$, $d_2 = c_2$, $d_3 = c_3$, and $d_4 = -c_4$.

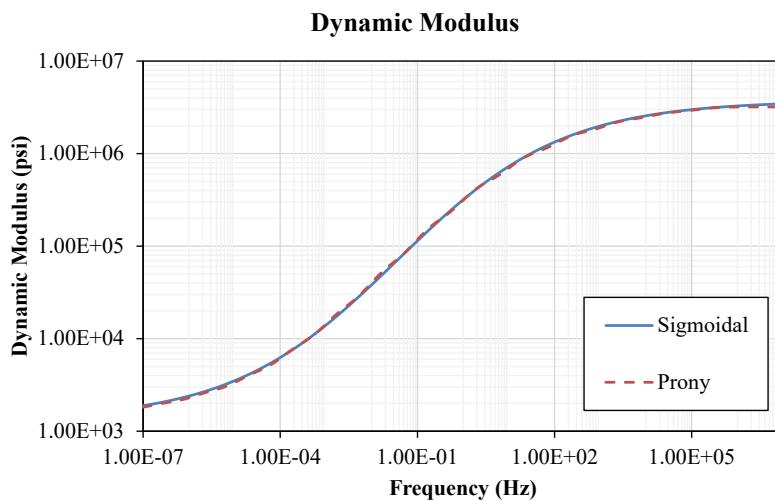


Figure 16. Dynamic Modulus Curves from Prony Series and Sigmoidal Function Using $d_1 = c_1$, $d_2 = c_2$, $d_4 = -c_4$, and d_3 Determined from NLS.

Conversion from Dynamic Modulus (Frequency Dependent) to Relaxation Modulus (Time Dependent)

Interconversion from dynamic modulus to relaxation modulus is slightly more involved. The procedure used in ViscoWave is described as the following.

1. Since the sigmoidal coefficients for the dynamic modulus (d_1 through d_4 of Equation (76)) are known, it is necessary to determine the Prony coefficients by fitting Equation (82) to Equation (76) by means of NLS.

However, it is important that the NLS begins with a reasonable initial guess for the Prony coefficients (i.e., E_0 and E_i values) in Equation (82). To find the initial guess, the relaxation modulus in Equation (58) is constructed with the values of c_1 , c_2 , and c_4 determined from Equations (83-1), (83-2), and (86-2), respectively. **For the unknown c_3 value, assume it is equal to d_3 (for now).** Then run the LLS to determine the Prony coefficients in Equation (59). These Prony coefficients become the initial guesses for the NLS of Equation (82). Figure 17 shows how this Prony series dynamic modulus compares to the original sigmoidal function.

2. Using the above initial guesses for E_0 and E_i values, run NLS to determine the final E_0 and E_i values that minimizes the error between the dynamic modulus curves from Prony series and sigmoidal function. Figure 18 shows a comparison between the post-NLS Prony series function and the original sigmoidal function.

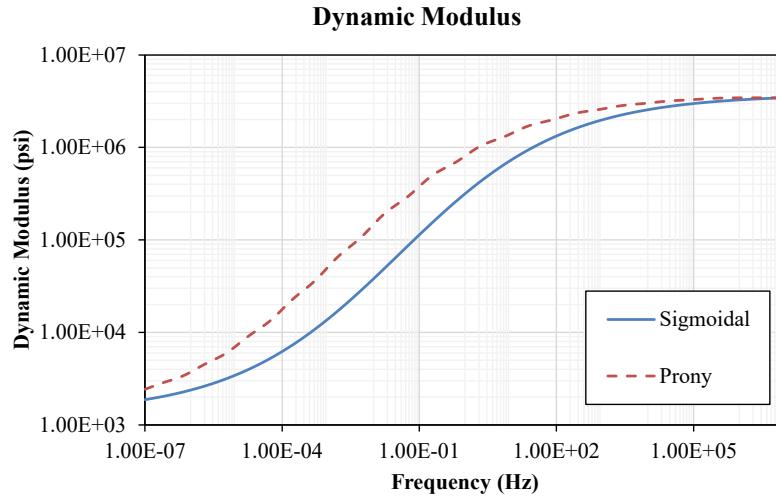


Figure 17. Dynamic Modulus Curves from Original Sigmoidal Function and Prony Series with Coefficients Determined from LLS.

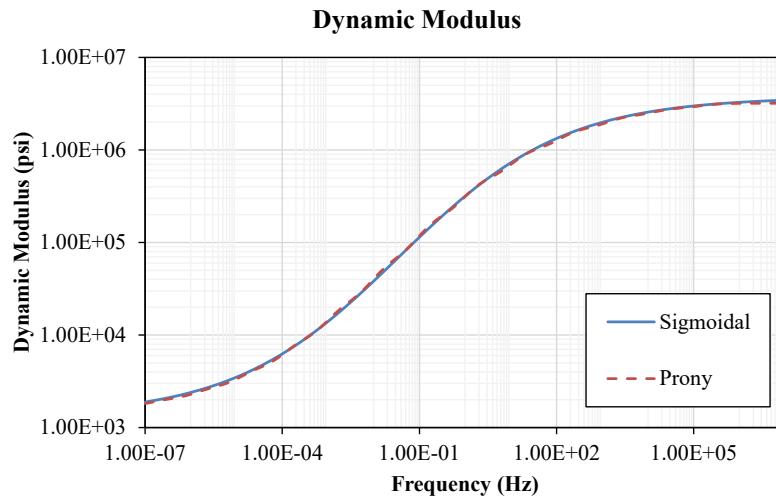


Figure 18. Dynamic Modulus Curves from Original Sigmoidal Function and Prony Series with Coefficients Determined after NLS.

3. Reconstruct the relaxation modulus in Equation (58) with the values of c_1 , c_2 , and c_4 determined from Equations (83-1), (83-2), and (86-2), respectively. **For the unknown c_3 value, assume it is equal to d_3 (for now).** Then run the NLS again to determine the c_3 value, by minimizing the error between the Prony series relaxation modulus and the sigmoidal function. The final outcome looks identical to the curves shown previously in Figure 13.

 If you have dynamic modulus data from laboratory testing and would like to use ViscoWave for simulation, you need to use the above procedure for converting the frequency dependent dynamic modulus to time dependent relaxation modulus.

The above procedures have been implemented into the “Dynamic_Modulus_Calc” worksheet of ViscoWave spreadsheet.

TIME-TEMPERATURE SUPERPOSITION AND MODULUS MASTER CURVE

The constitutive relations (i.e., relaxation or dynamic modulus) discussed so far did not account for the effect of temperature (since these properties were assumed to be function of time or frequency only). Hence, they are only adequate when the material is subjected to a constant temperature (in this context, the temperature of AC at the time of FWD testing). In order to account for the effect of temperature, it is necessary to generalize the relaxation or dynamic modulus values described above to include temperature as an independent variable. For example, the relaxation modulus in time domain can now be defined as the following.

$$E = E(t, T) \quad (87)$$

where the uppercase T has been used for temperature to distinguish from the time variable, t .

Fortunately, there has been sufficient theoretical and experimental evidence showing that most linearly viscoelastic materials obey the Time-Temperature Superposition Principle (TTSP) which allows for combining the effect of time and temperature (Findley et. al., 1976, Tschoegl, 1989, Wineman and Rajagopal, 2000). A viscoelastic material obeying the TTSP is also frequently referred to as a “Thermorheologically Simple” material. According to the TTSP, the above relaxation modulus can be written as the following:

$$E = E(t, T) = E(\xi, T_R) \quad (88)$$

where T_R is the reference temperature and ξ is called the “Reduced Time” which is related to the physical time in the following manner:

$$\xi = t/a_T(T) \quad (89)$$

where a_T is the shift factor. The above equation can be also be written as:

$$\log(\xi) = \log(t/a_T(T)) = \log(t) - \log(a_T(T)) \quad (90)$$

The above equations indicate that fundamental property of a viscoelastic material at the time t and at temperature T is equal to that property at the reduced time ξ and at the reference temperature T_0 . This in turn, indicates that the fundamental properties at different temperatures can be shifted by the amount $\log(a_T)$ when plotted in log-scale (or stretched/shrunk in arithmetic-scale), as shown conceptually in Figure 19 for the construction of dynamic modulus “Mastercurve” from laboratory test data.

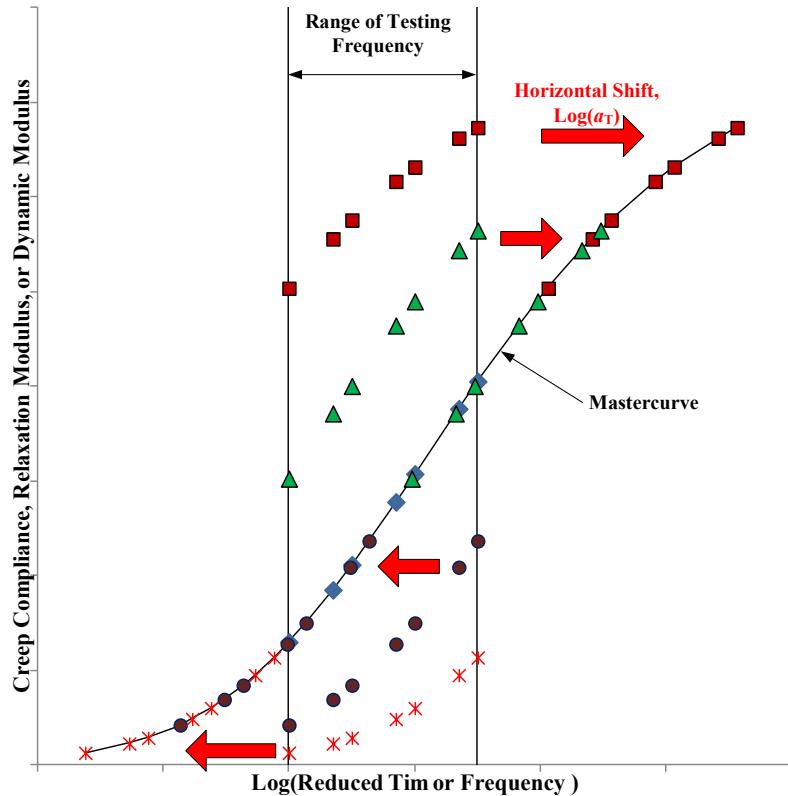


Figure 19. Concept of Time-Temperature Superposition Principle and Mastercurve

❑ The laboratory equipment for dynamic modulus testing is typically limited in the range of loading frequencies and temperature. As such, the usual practice is to test the AC material at the limited frequency ranges but at different temperatures for constructing the dynamic modulus master curve.

Unlike the laboratory modulus testing that can be conducted at multiple temperatures, the FWD data often shows further limitations due to the challenge associated with obtaining the data at a wide range of temperatures (e.g., FWD testing conducted at the same location in different seasons).

Due to the practical efficiency issue of FWD testing at multiple temperatures in the field, a typical Time-Temperature shift factor equation is adopted in ViscoWave. The shift factor equation takes the quadratic form shown below.

$$\log(a_T) = aT^2 + bT + c \quad (91)$$

At the reference temperature (i.e., $T = T_R$), the shift factor a_T must equal to 1.0 (i.e., $\xi = t$ in Equation (88) and Equation (89)), which leads to $\log[a_T(T_R)] = 0$. Upon substituting this into the above equation, one obtains the following for the constant c .

$$c = -aT_R^2 - bT_R \quad (92)$$

Substituting Equation (92) back into Equation (91) results in the following equation that is implemented into ViscoWave spreadsheet.

$$\log(a_T) = a(T^2 - T_R^2) + b(T - T_R) \quad (93)$$

 According to Sakhaefar (2011), the Time-Temperature shift factor equations from a variety of AC mixtures in North Carolina did not show a significant variability. As such, the default values for a , b , and c in Equation (91) are set to 0.0007, -0.1592, and 3.0489, respectively. These values were obtained from the average shift factor equation reported by Sakhaefar (2011).

CHAPTER 4 – VISCOWAVE USER GUIDE

LICENSE AGREEMENT

ViscoWave is an open-source program that is made available to you at no cost, under the GNU Affero General Public License V3 (<https://www.gnu.org/licenses/agpl-3.0.en.html>).

INSTALLING VISCOWAVE AND PORTABLE R

ViscoWave installer and all its source code are available at the GitHub open-source repository (<https://github.com/leehyu20/ViscoWave>), as shown in Figure 20.

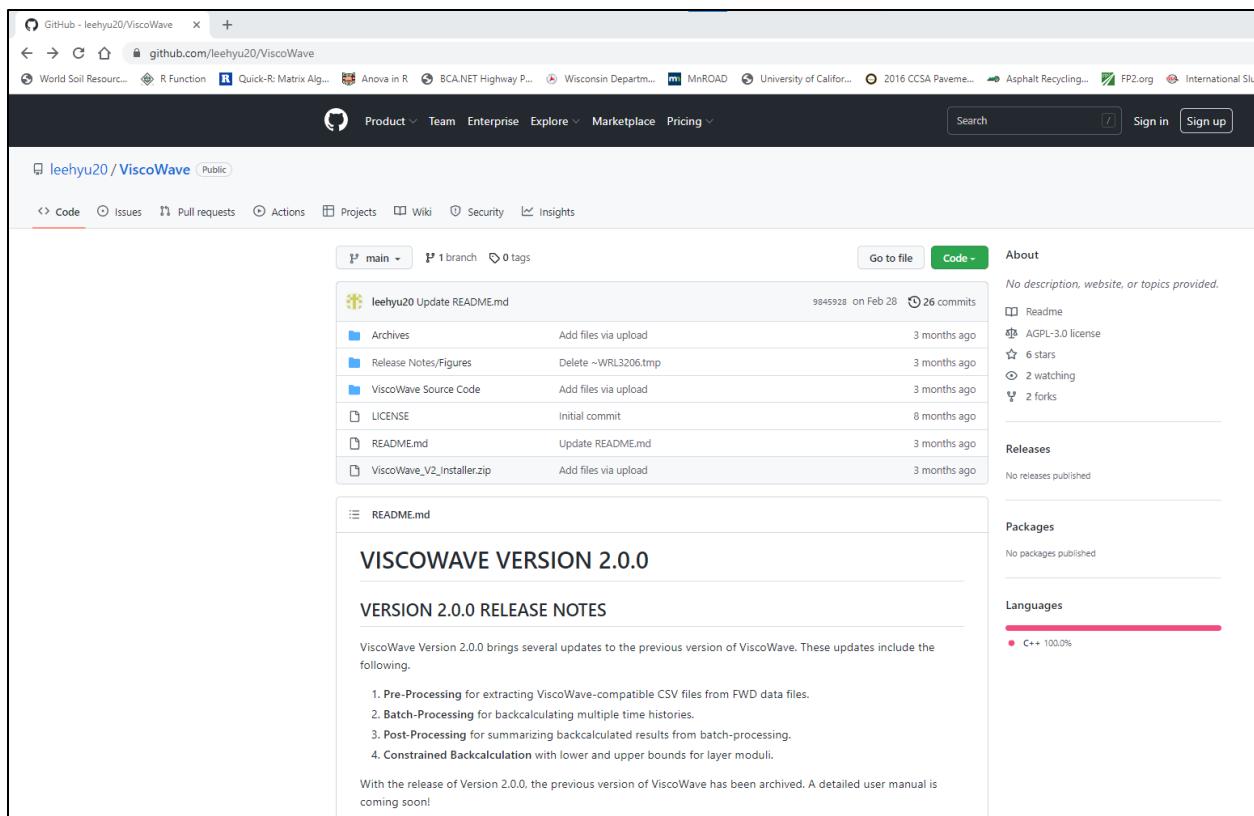


Figure 20. GitHub Open-Source Repository for ViscoWave

ViscoWave installer can be downloaded by clicking on the link to the zip file, or directly from this link: [ViscoWave_V2_Installer.zip](#).

To install ViscoWave, extract the files from the .zip file and save them to a local directory. Then double click the Setup.exe file. Follow the on-screen instructions to complete installation, as shown in Figure 21. Once the installation is complete, you will see a shortcut to the ViscoWave template on your desktop.

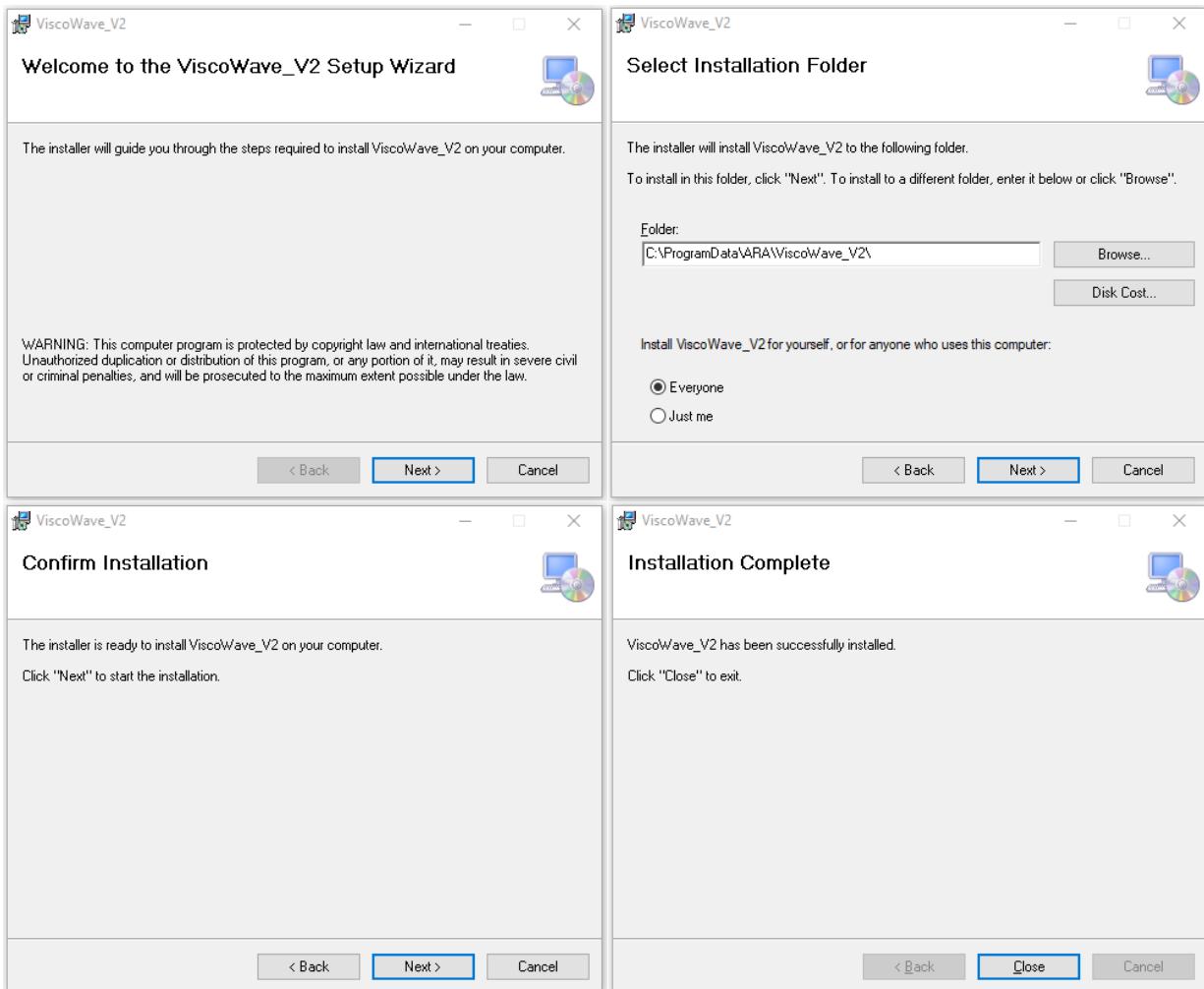


Figure 21. Installing ViscoWave

■ Installation of ViscoWave requires that the user has **Administrator Access** to the computer. ViscoWave (and its installer) has only been tested on **64-bit Windows 10 and Windows 11 Environment, with 32-bit or 64-bit MS Excel**.

In addition to the above, it is strongly recommended that you also download and execute “[Portable R for VW.msi](#)” to install **Portable R** in your machine. Portable R is needed for Pre-Processing of the FWD data (i.e., extracting time history CSV files).

The Portable R must be installed under the following path of your Windows machine:

- C:\ProgramData\R Portable for ViscoWave

■ The advantage of using the above msi file is that it installs Portable R and the necessary packages into the designated path. Therefore, it is strongly recommended that you use the above msi file for installing Portable R.

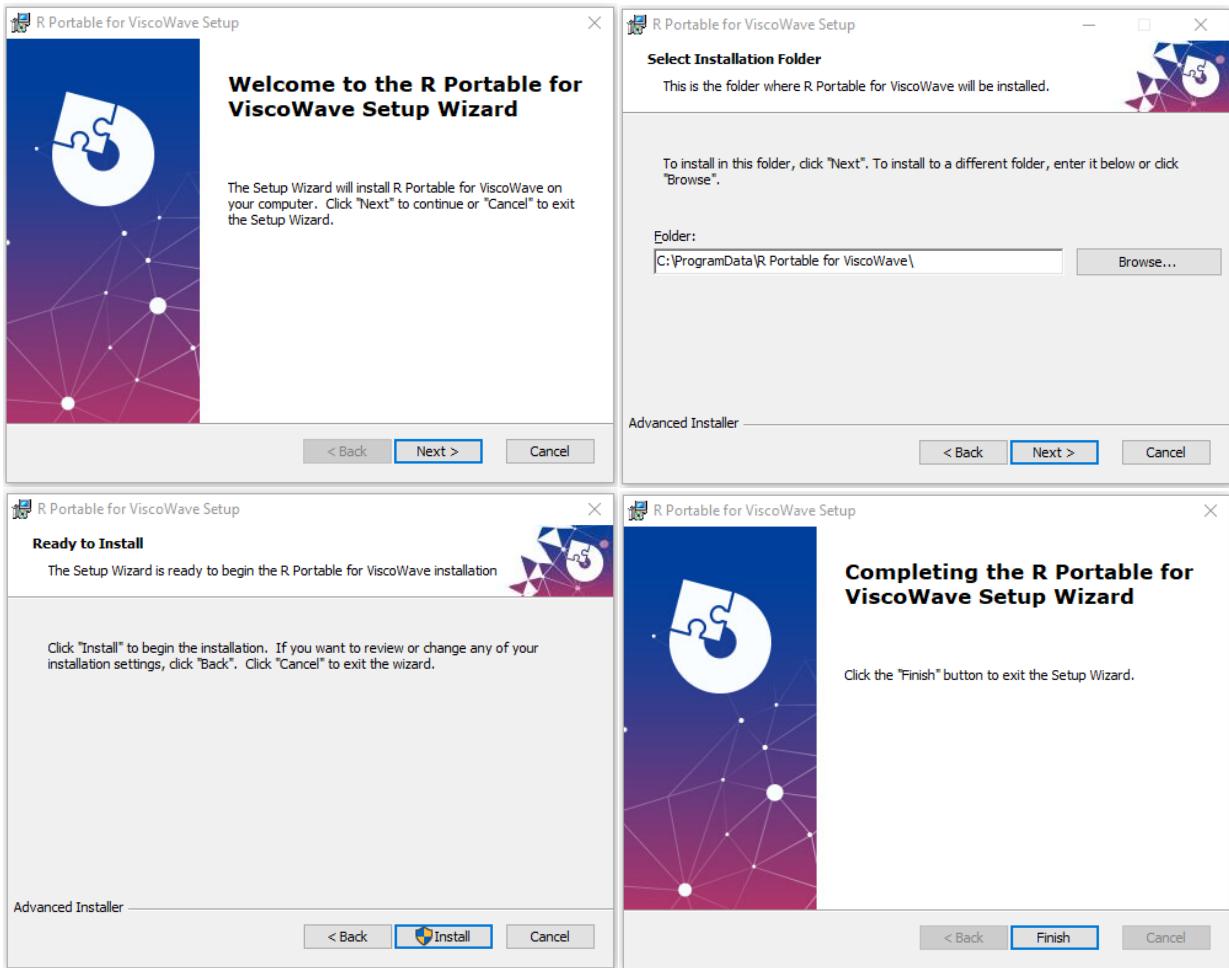


Figure 22. Installing Portable R for ViscoWave

However, it is also possible to install Portable R by downloading it from SourceForge (<https://sourceforge.net/projects/rportable/>) and extracting the files to the path shown above. In this case, the necessary R packages may need to be installed manually. To do so, start R console by double clicking the “R.exe” file located in the following directory, depending on the version of MS Excel you are using.

- C:\ProgramData\R Portable for ViscoWave\App\R-Portable\bin\R.exe (If you are using **32-bit** version of Excel)
- C:\ProgramData\R Portable for ViscoWave\App\R-Portable\bin\i386\R.exe (If you are using **64-bit** version of Excel)

Once the R terminal pops up on your screen, type in the following command to install the necessary packages (Figure 23).

```
install.packages(c('data.table', 'plyr', 'dplyr', 'stringr', 'gsubfn', 'RODBC', 'svDialogs', 'ggplot2'))
```

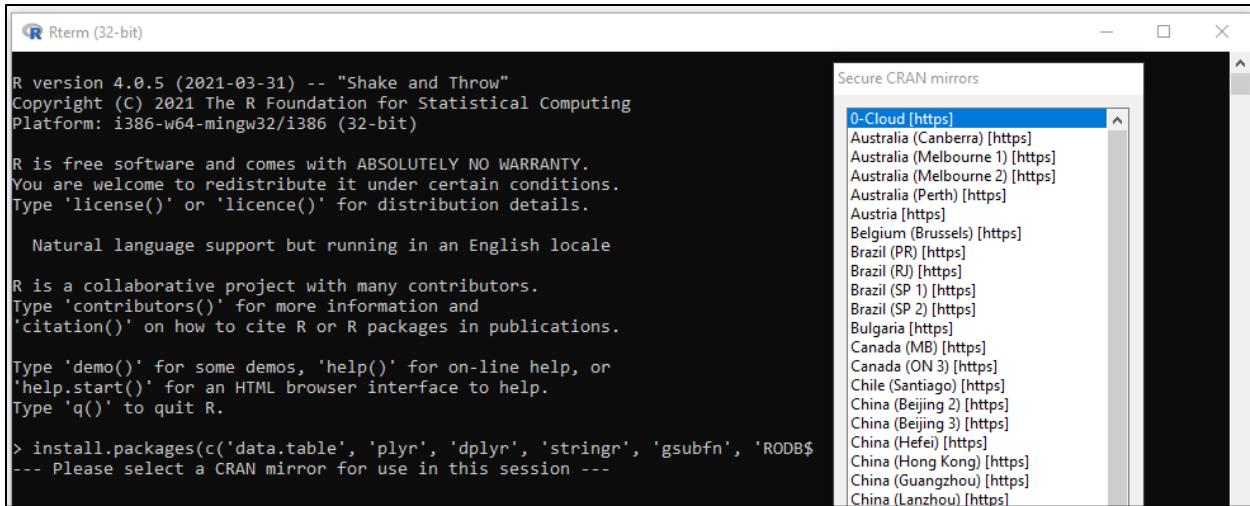


Figure 23. Manually Installing Packages to Portable R

If you do not wish to install Portable R, you will have to run the R-script generated by ViscoWave in Standard R (with or without RStudio), outside the ViscoWave environment. This option is not recommended unless you have sufficient knowledge in R language.

BEFORE GETTING STARTED

ViscoWave interface is implemented in a macro-enabled MS Excel spreadsheet environment. As such, it is important that the macros or code written in Visual Basic for Application (VBA) language be enabled for the interface to function properly.

Figure 23 shows the Excel window in which a security warning is displayed due to the embedded macros. If you see such a message, simply click the “Enable Content” button to enable the macros within the ViscoWave interface.

Furthermore, it is recommended that the user enables all macros within the Excel environment. To do so, follow the steps outlined below.

1. Click on the “Developer” tab of the Excel Ribbon
 - a. If the “Developer” tab is not visible, go to File → Options → Customize Ribbon, and select the check box next to “Developer”.
2. Select “Macro Security” in the “Developer” tab
3. In the new pop-up window, select “Enable All Macros” option and then click “OK” (Figure 24).

With the above settings in Excel, you are ready to run ViscoWave! The remainder of this manual describes how to use ViscoWave for simulation and backcalculation.

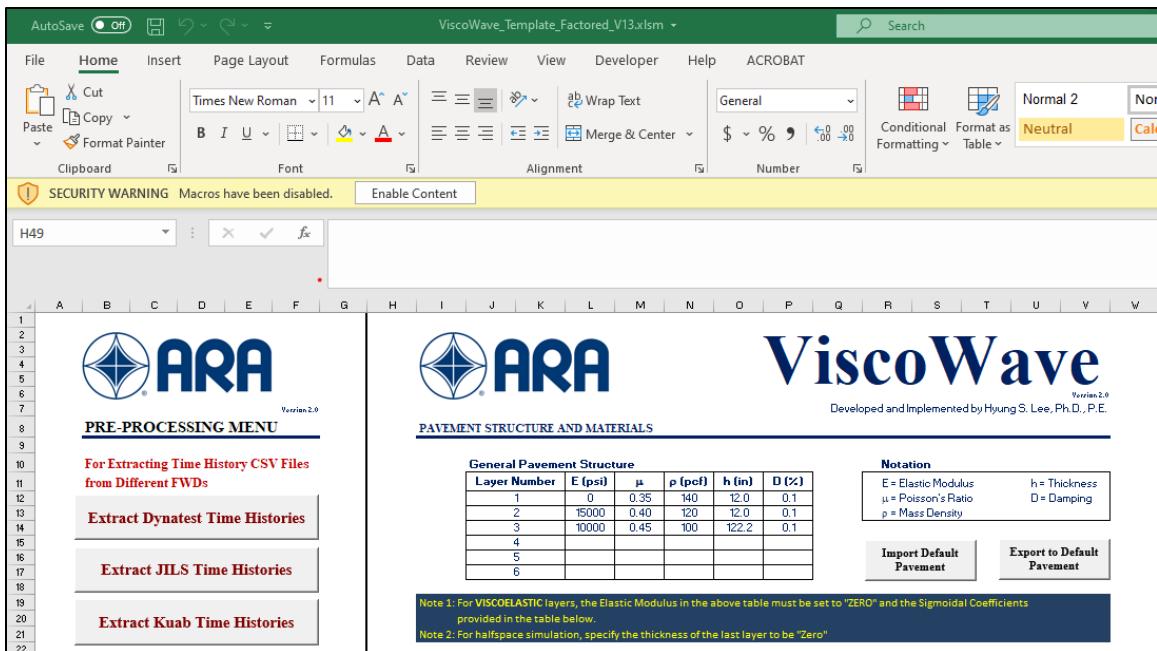


Figure 23. Security warning message in Excel

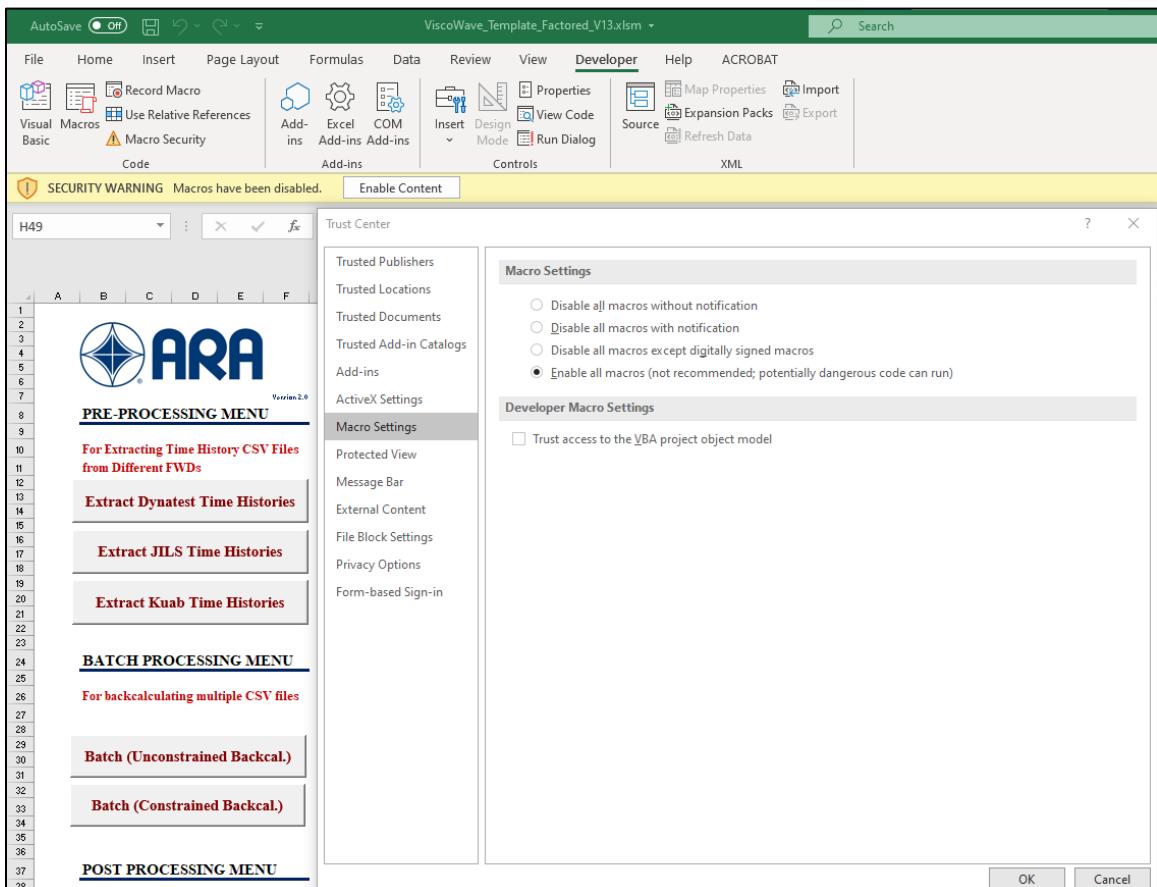


Figure 24. Enabling all macros in Excel

GETTING STARTED

Once ViscoWave is installed, you should see a ViscoWave shortcut created on your Windows Desktop. Double-click on the shortcut to open the ViscoWave interface built in the Microsoft Excel environment.

 The desktop shortcut opens the ViscoWave interface which is a “**Read-Only**” file. Please save the template with a different file name if you do not want to lose your changes.

ViscoWave Excel interface has the following 5 tabs (that are visible by default):

- **VW** – Tab for ViscoWave main menu as well as forward simulation and backcalculation.
- **Default Inputs** – Tab for storing the default (or seed) pavement parameters (Modulus, Poisson’s Ratio, Density, Thickness, and Damping) and their constraints.
- **Dynamic_Modulus_Calc** – Tab for converting the Relaxation Modulus (in time-domain) to Dynamic Modulus (in frequency-domain), or vice versa.
- **VW_Quick_Start_Guide** – Tab including a quick reference for using ViscoWave.
- **AGPL License V3** – Tab showing the license agreement (AGPL Version 3.0) for ViscoWave.

Among the above, you will mostly interact with “VW” and “Dynamic_Modulus_Calc” tabs for the analysis as described subsequently. In addition to the above, the interface includes the following 2 tabs that are hidden:

- **DM_Summary** – Tab for summarizing the backcalculated Dynamic Modulus (i.e., for Viscoelastic Layers only).
- **ElasticLayer_Summary** – Tab for summarizing the backcalculated Elastic Modulus (i.e., for Elastic Layers).

These 2 tabs are only reserved for summarizing the backcalculated results from batch-mode processing. As a user, you do not need to worry about these tabs.

VISCOWAVE ANALYSIS MENU OPTIONS

Figure 25 shows the ViscoWave Main Menu options that can be found within the VW tab. Additional details regarding these menu options are provided in the subsequent sections.

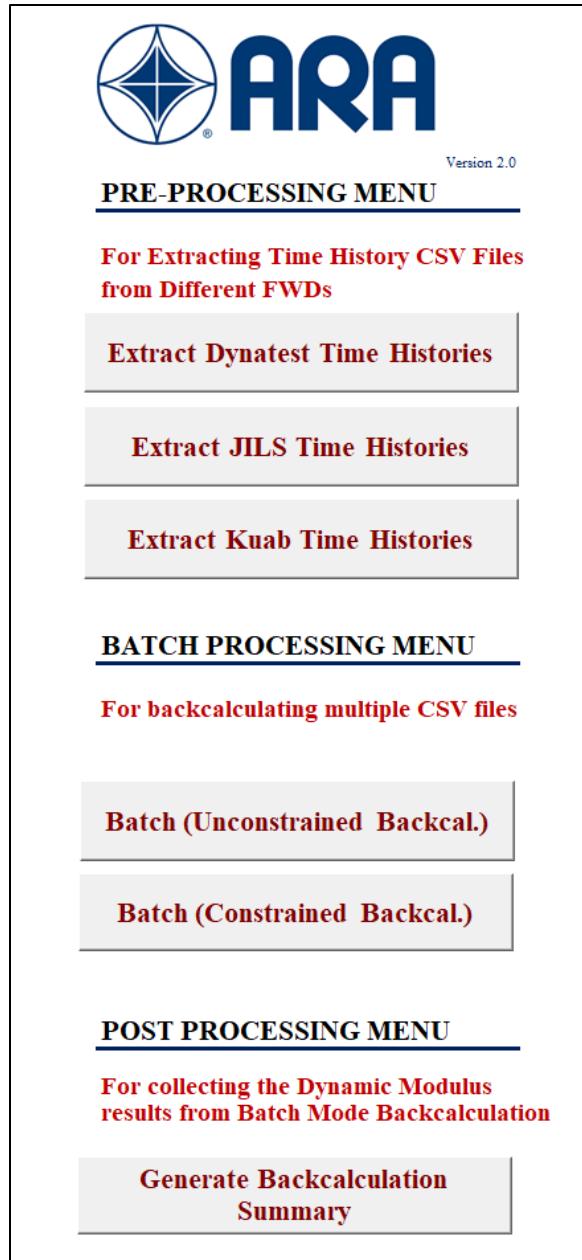


Figure 25. ViscoWave Analysis Options (in VW Tab)

PRE-PROCESSING OF FWD DATA

Pre-processing of the FWD data allows you to extract deflection summaries as well as ViscoWave-compatible time history CSV files. Samples of Dynatest, JILS, and Kuab FWD files are provided in [Dropbox](#).

To extract the CSV files, you need to select one of the three FWD types (Dynatest, JILS, and Kuab FWDs). Once the appropriate button is clicked, the program will ask you to select the

FWD data for pre-processing. The compatible FWD files and folder structures are described below.

Extracting Time History Data from Dynatest FWD Files

The Dynatest FWD data (peak deflections, time histories, temperature, etc.) is stored in ***.mdb** file format. Click on the “Extract Dynatest Time Histories” button and select the ***.mdb** file when prompted, as shown in Figure 26.

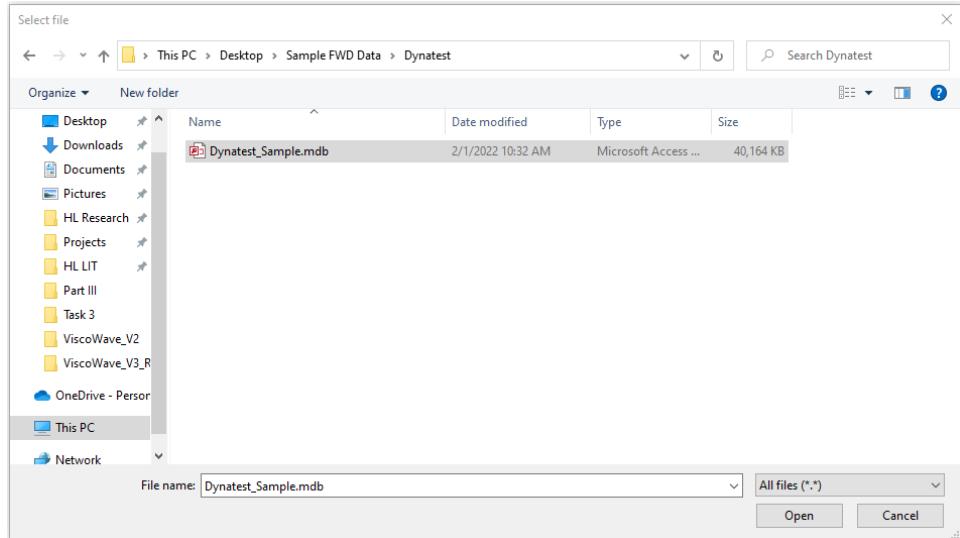


Figure 26. Selecting Dynatest File for Extraction

Older Dynatest FWDs may not produce *.mdb files. Instead, these older models save the time histories in *.fwd or other ASCII file formats. These older files are not supported by ViscoWave at this time.

If you chose not to install Portable R, you need to run the R-script in Standard R. The script is saved in “Dynatest.r” under the same path as the ViscoWave interface you are running.

Extracting Time History Data from JILS FWD Files

For JILS FWD, the peak deflections are saved in the ***.DAT** file while the time histories for all drops are stored in ***.THY** file. Both these files should have the same name (e.g., JILS.DAT and JILS.THY) and be placed in the same path. You may select either file for pre-processing, as shown in Figure 27.

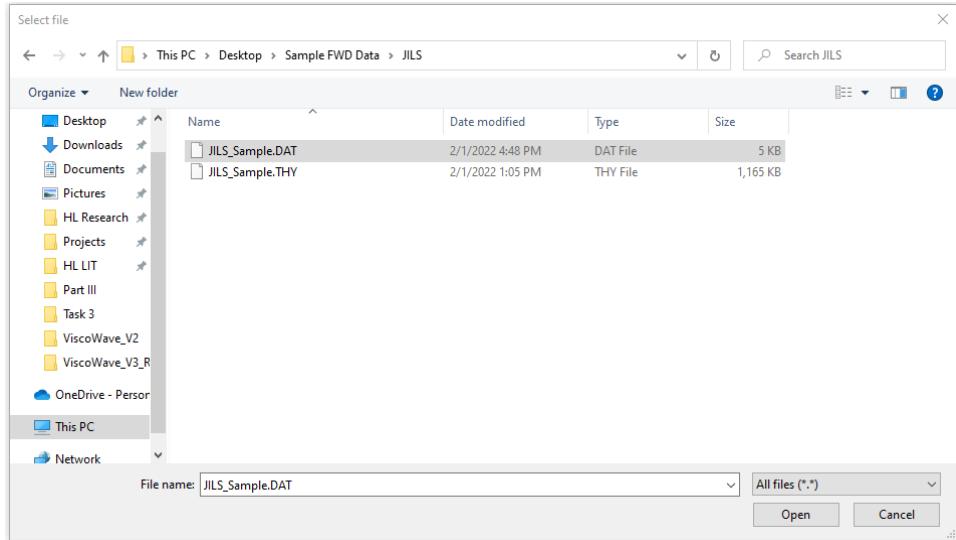


Figure 27. Selecting JILS File for Extraction

☞ If you chose not to install Portable R, you need to run the R-script in Standard R. The script is saved in “JILS.r” under the same path as the ViscoWave interface you are running.

Extracting Time History Data from Kuab FWD Files

For Kuab FWD, the peak deflections are saved in the ***.fwd file** while the time histories for each drop are stored in ***.HST file** (i.e., there are multiple *.HST files corresponding to one *.fwd file). All these files need to be placed in the same path, as shown in Figure 28. When prompted by the program, select the folder that contains all these files (Figure 29).

☞ If you chose not to install Portable R, you need to run the R-script in Standard R. The script is saved in “Kuab.r” under the same path as the ViscoWave interface you are running.

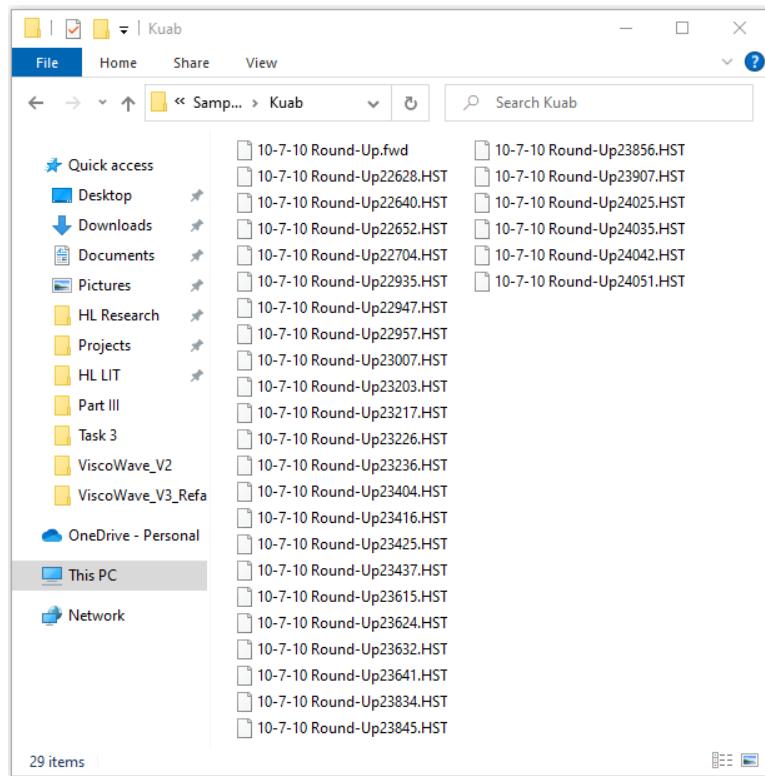


Figure 28. Kuab FWD and HST Files

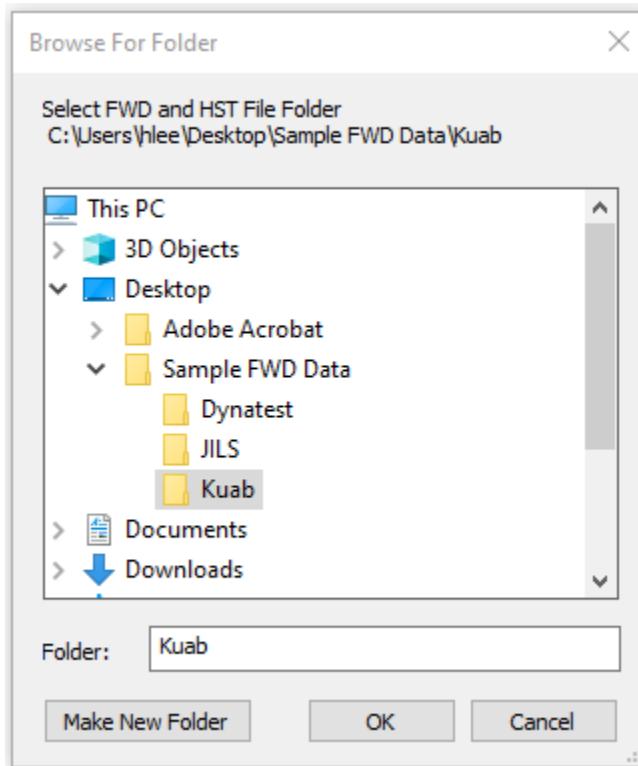


Figure 29. Selecting Kuab Folder Location for Data Extraction

File Structure for Extracted Time History Data

The pre-processing module generates multiple files, depending on the FWD data. Figure 30 shows the structure of the files and folders produced from pre-processing. For Dynatest and JILS data, these new files are stored in a new folder created with the same name as the FWD data file. This folder is regarded as the root directory in the file structure. For Kuab data, the root directory is the folder that included the original FWD data.

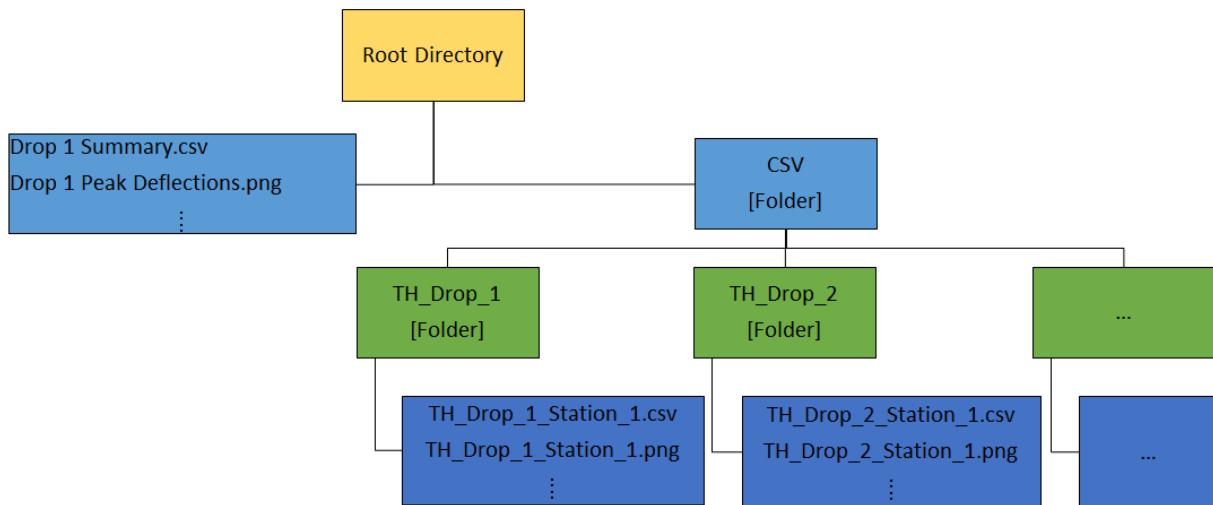


Figure 30. Pre-Processed File Structure

The root directory contains several CSV files (one CSV file per FWD drop number) that summarizes the peak load and deflections along with other information stored in the FWD files (e.g., station, temperature, drop number, etc.). In addition, the peak FWD deflection plots are also saved in the root directory in *.png format.

Figure 31 shows an example of the files stored in the root directory and Figure 32 shows an example of the peak deflection plot.

Although the peak deflections are not used for dynamic backcalculation, it is always good practice to investigate the peak deflections for identifying any outliers or non-decreasing deflections to be excluded from the analysis, and/or for determining any trends within the data.

File Home Share View Manage Dynatest_Sample

Quick access

	Name	Date modified	Type	Size
CSV	6/23/2022 10:26 AM	File folder		
Drop_1_Peak Deflections.png	2/24/2022 12:17 PM	PNG File	55 KB	
Drop_1_Summary.csv	2/24/2022 12:17 PM	Microsoft Excel C...	11 KB	
Drop_2_Peak Deflections.png	2/24/2022 12:17 PM	PNG File	55 KB	
Drop_2_Summary.csv	2/24/2022 12:17 PM	Microsoft Excel C...	11 KB	
Drop_3_Peak Deflections.png	2/24/2022 12:17 PM	PNG File	56 KB	
Drop_3_Summary.csv	2/24/2022 12:17 PM	Microsoft Excel C...	11 KB	
Drop_4_Peak Deflections.png	2/24/2022 12:17 PM	PNG File	57 KB	
Drop_4_Summary.csv	2/24/2022 12:17 PM	Microsoft Excel C...	11 KB	
Drop_5_Peak Deflections.png	2/24/2022 12:17 PM	PNG File	57 KB	
Drop_5_Summary.csv	2/24/2022 12:17 PM	Microsoft Excel C...	11 KB	
Drop_6_Peak Deflections.png	2/24/2022 12:17 PM	PNG File	57 KB	
Drop_6_Summary.csv	2/24/2022 12:17 PM	Microsoft Excel C...	11 KB	

13 items | 1 item selected 54.5 KB

Figure 31. Files Created in Root Directory from ViscoWave Pre-Processing

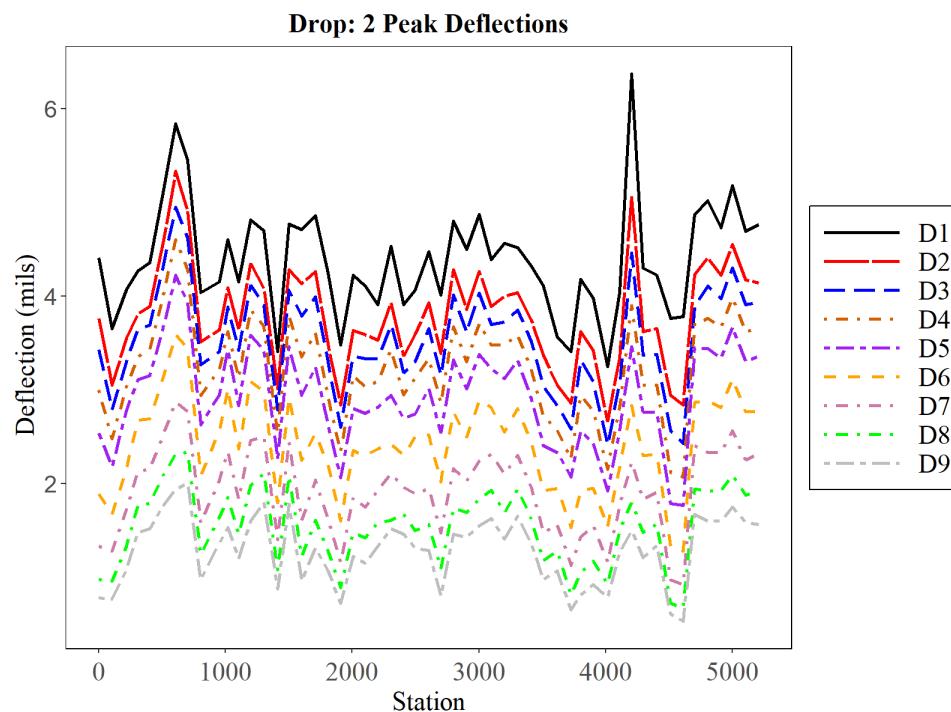


Figure 32. Example Peak Deflection Plot

The root directory also contains another folder named “CSV”. In this folder, there are several subfolders (one for each FWD drop number) with the CSV files including the FWD load and deflection time-history data (Figure 33). These CSV files are compatible with ViscoWave and can be inputted directly into the ViscoWave interface for simulation or backcalculation.

The load and deflection time history plots are also saved in the subfolder, in *.png format (Figure 34).

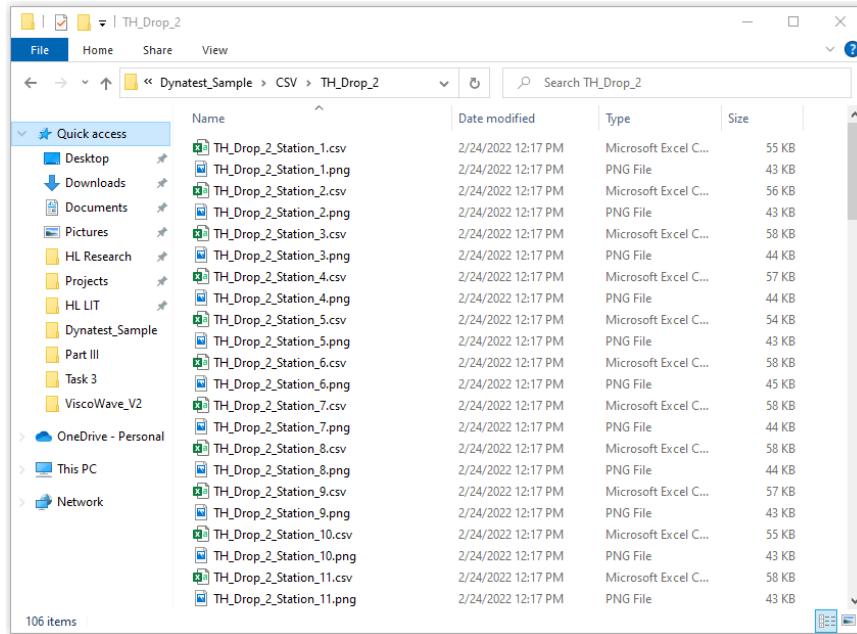


Figure 33 Time History Files Created from ViscoWave Pre-Processing

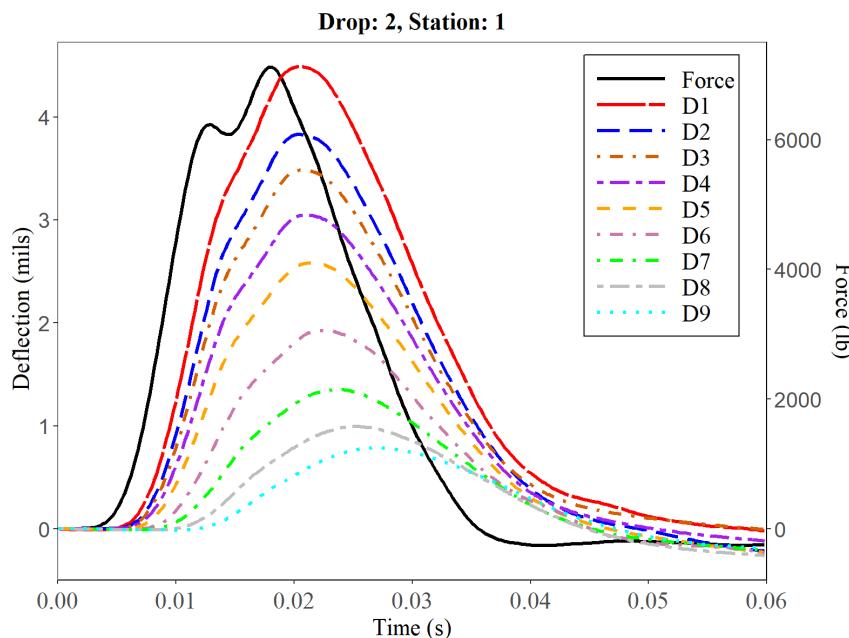


Figure 33. Example Time History Plot

 It is recommended that you go through the time history plots to ensure they are smooth, and no issues or outliers are found (at least visually). You can easily go through these figures in any image viewing app (e.g., Windows application Photos).

Figure 34 shows an example of a “good” versus “bad” time history. Do not use any of these “bad” looking time histories for backcalculation.

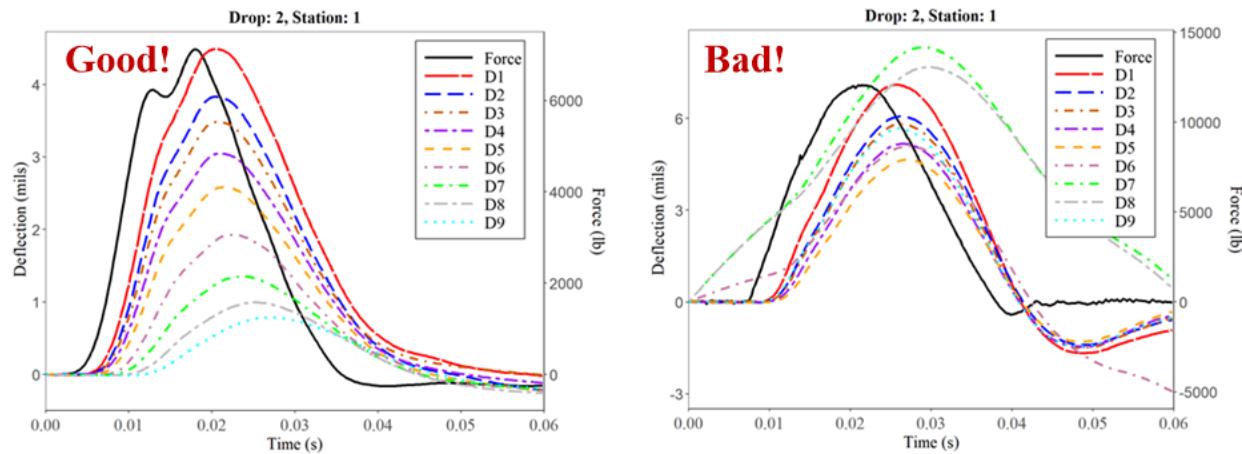


Figure 34. Good and Bad Examples of FWD Time History Data

FORWARD SIMULATION USING VISCOWAVE

Forward simulation can be run entirely in the VW tab of the ViscoWave interface. The minimum inputs needed for the forward simulation include the following.

- **General Pavement Structure** information (Figure 35), including the modulus (E), Poisson’s ratio (μ), unit weight (ρ), thickness (h), and damping (D).
 - If you want to include 1 or more viscoelastic layers, set the modulus that particular layer to zero in the “General Pavement Structure” table, and provide the coefficients for the **relaxation modulus** in the table below it.
- Locations for the **FWD Deflection Sensors** in inches (Figure 35). ViscoWave currently allows up to 9 deflection sensors.
- **Radius of the Load Plate** in inches (Figure 35). You do **NOT** need to fill in the peak load and pressure, as these parameters will be calculated from the load time history, as described below.
- **Load (or Force) Time History** in the bottom table (Figure 36). You may import a CSV file (generated from pre-processing) if you want to use the time history measured by an actual FWD. Otherwise, the interface also allows you to generate a few synthetic load time histories: **Half-Sine, Haver-Sine, and Normal (Gaussian) shaped loads**. Simply

fill in the necessary inputs for the load you want then click on the corresponding button to generate the load.


ViscoWave
 Version 2.0
Developed and Implemented by Hyung S. Lee, Ph.D., P.E.

PAVEMENT STRUCTURE AND MATERIALS

General Pavement Structure					
Layer Number	E (psi)	μ	ρ (pcf)	h (in)	D (%)
1	0	0.35	140	12.0	0.1
2	15000	0.40	120	12.0	0.1
3	10000	0.45	100	122.2	0.1
4					
5					
6					

Note 1: For VISCOELASTIC layers, the Elastic Modulus in the above table must be set to "ZERO" and the Sigmoidal Coefficients provided in the table below.

Note 2: For halfspace simulation, specify the thickness of the last layer to be "Zero".

Sigmoidal Coefficients (Relaxation)					
Layer Number	c1	c2	c3	c4	T (°F)
1	3.123	3.446	-0.128	0.554	68.0
2					
3					

Relaxation Modulus Function:

$$\log E(t) = c_1 + \frac{c_2}{1 + e^{c_3 + c_4 \log(t_r)}}$$

FWD DEFLECTION SENSORS

Sensor Location (in.)									
D1	D2	D3	D4	D5	D6	D7	D8	D9	
0	8	12	18	24	36	48	60		

FWD LOAD SUMMARY

Load Radius: in.
Peak Load: lbs
Peak Pressure: psi

Note 1: Only enter Load Radius

Note 2: Do NOT enter Peak Load and Pressure. These values are obtained from the Load Time History entered below

Figure 35. ViscoWave Main Screen

LOAD AND DEFLECTION TIME HISTORIES

Note 1: Load Time History must be inputted in the 2nd column highlighted in ORANGE

Note 2: Time Increment (dt) must be 0.2 milliseconds (Hard Coded in C++ Code)

Note 3: Maximum time should not exceed 0.07 seconds. It is generally recommended not to modify the time column.

Note 4: Do NOT modify the Normalized Load History in Column P

Use Synthetic Load Time History

Half-Sine Load		
Begin:	<input type="text" value="0.005"/>	sec.
End:	<input type="text" value="0.03"/>	sec.
Magnitude	<input type="text" value="9000"/>	lb
Generate Half-Sine Load		

HaverSine Load		
Begin:	<input type="text" value="0.005"/>	sec.
End:	<input type="text" value="0.03"/>	sec.
Magnitude	<input type="text" value="9000"/>	lb
Generate Haver-Sine Load		

Normal Distribution (Gaussian)		
Centered	<input type="text" value="0.018"/>	sec.
Std. Dev.	<input type="text" value="0.004"/>	sec.
Magnitude	<input type="text" value="9000"/>	lb
Generate Gaussian Load		

Measured Deflections

Time	Force	D1	D2	D3	D4	D5	D6	D7	D8	D9	Temperature
0	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	68.00
0.0002	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	68.00
0.0004	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	68.00
0.0006	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	68.00
0.0008	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	68.00
0.001	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.00	68.00
0.0012	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.00	68.00
0.0014	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.00	68.00
0.0016	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	68.00

Clear Measured Time History

Import Measured Time History

Figure 36. Time History Table in VW Tab

Once the above inputs are provided in the interface, the simulation results will automatically be updated, and the time history plot(s) will be updated, as shown in Figure 37. Note that, if you used a synthetic load (i.e., without importing measured FWD data), the measured deflection time history plot will not show any data.

A few important notes:

1. To simulate a **viscoelastic layer**, the modulus of that particular layer must be assigned a value of ZERO in the General Pavement Structure table, and the sigmoidal function coefficients should be provided in the table below it.
2. To simulate a **half-space (i.e., a semi-infinite subgrade)**, the thickness of the last layer must be specified to ZERO. However, it is generally recommended **NOT to use a half-space for backcalculation purposes** – use a very thick subgrade instead (e.g., 300 in. to 500 in. subgrade thickness).
3. Current ViscoWave interface allows up to 6 pavement layers (including 3 viscoelastic layers) and 9 deflection sensors. This is simply due to the interface, not the limitation of ViscoWave code.
4. **Time Increment (dt)** must be **0.2 milliseconds** (Hard Coded in C++ Code), and **maximum time** should NOT exceed 0.06 seconds. It is generally recommended not to modify the time column. In other words, use the time from 0 to 0.0598 sec. at an interval of 0.002 sec.

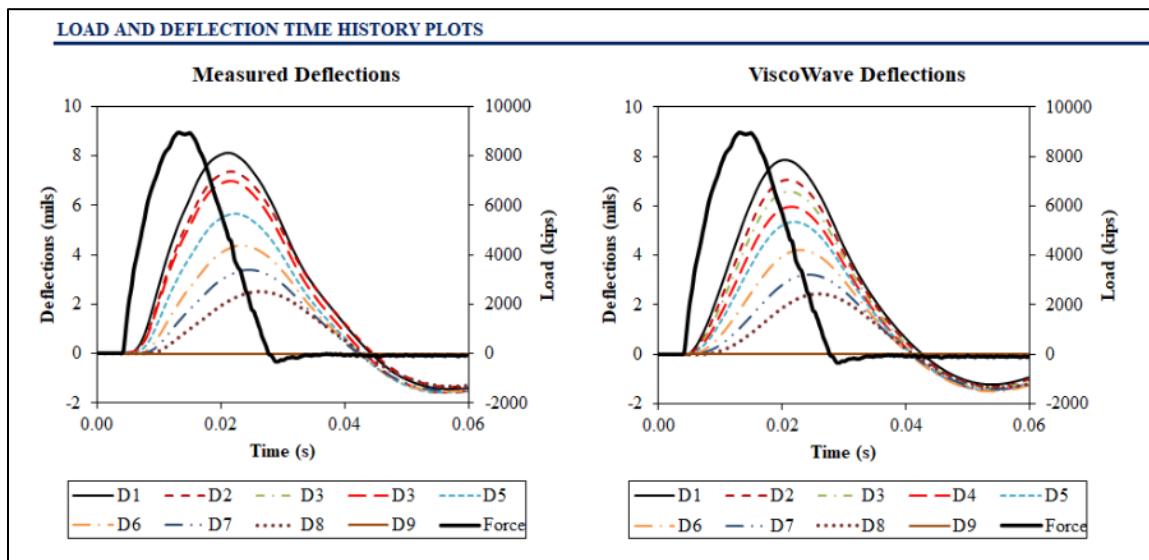


Figure 37. Measured and Simulated Time History Plots in VW Tab

BACKCALCULATION USING VISCOWAVE (SINGLE DROP)

Backcalculation of a single FWD time history can also be done entirely within the "VW" worksheet (Assuming the CSV files were extracted from the FWD data using the pre-processing options).

To backcalculate the pavement properties from a single CSV file, follow these steps:

1. Import the CSV file by clicking on the "Import Measured Time History" button and selecting the CSV file to be analyzed (Figure 36).
2. Fill out the pavement/materials related input (Figure 35), until there is a reasonable agreement between the measured and simulated time histories.
3. Fill out the constraints if constrained backcalculation is intended (Figure 38).
4. Try backcalculating the layer moduli (Unconstrained or Constrained, as desired).
5. If the backcalculated results are not satisfactory, try backcalculating with a different set of seed values.
6. If the backcalculated results are satisfactory, these results can be exported to the "Default Inputs" worksheet (if desired) by clicking on the appropriate button shown in Figure 35.

<u>BACKCALCULATION (UNCONSTRAINED OR CONSTRAINED)</u>												
Unconstrained Backcalculation						Constrained Backcalculation (Specify Upper and Lower Bounds in the Tables Below)						
Run Backcalculation (Unconstrained)						Run Backcalculation (Constrained)						
Tables for Upper and Lower Bounds General Pavement Structure												
Layer Number	E (psi)		μ		ρ (pcf)		h (in)		D (%)			
	Min.	Max.	Min.	Max.	Min.	Max.	Min.	Max.	Min.	Max.		
	1	-	-	-	-	-	-	-	-	-	-	
	2	5000	35000	-	-	-	-	-	-	-	-	
	3	4000	15000	-	-	-	-	-	50.0	200.0	-	-
	4	-	-	-	-	-	-	-	-	-	-	
	5	-	-	-	-	-	-	-	-	-	-	
6	-	-	-	-	-	-	-	-	-	-		
Note: The cells for upper and lower bounds of Poisson's ratio, unit weight, and damping are reserved for future updates and advanced users												
Sigmoidal Coefficients (Relaxation)												
Layer Number	c1		c2		c3		c4					
	Min.	Max.	Min.	Max.	Min.	Max.	Min.	Max.				
	1	2.500	4.000	3.000	4.500	-	-	0.500	-			
	2	-	-	-	-	-	-	-	-			
3	-	-	-	-	-	-	-	-				

Figure 38. Options for Single File Backcalculation

❑ Note again that the Sigmoidal Function used or backcalculated in ViscoWave (in the VW tab) corresponds to the Relaxation Modulus in time-domain. **If you need to convert Relaxation Modulus to Dynamic Modulus (or Vice Versa), follow the instructions in the following section.**

Converting Relaxation Modulus to Dynamic Modulus

The “Dynamic_Modulus_Calc” tab shown in Figure 39 allows you to convert the Relaxation Modulus (in time-domain) to Dynamic modulus (in frequency-domain). Similarly, if you have Dynamic Modulus Sigmoidal Coefficients (say from laboratory testing of asphalt mixtures), and want to use them in ViscoWave, this tab also allows you to convert the Dynamic Modulus to Relaxation Modulus.

- To convert from **Relaxation Modulus ($E(t)$)** to **Dynamic Modulus ($|E^*|$)**, simply fill in the Relaxation coefficients (Top Left table) and click on the button “Convert $E(t)$ to $|E^*|$ ”.
- To convert from **Dynamic Modulus ($|E^*|$)** to **Relaxation Modulus ($E(t)$)**, simply fill in the Relaxation coefficients (Top Right table) and click on the button “Convert $|E^*|$ to $E(t)$ ”.

The tab also provides the Prony series coefficients that can be used to calculate both the Relaxation and Dynamic moduli, along with the plots for the moduli from both the Sigmoidal functions and Prony series (Figure 40).

Sigmoidal Coefficients [RELAXATION MODULUS, $E(t)$]				Sigmoidal Coefficients [DYNAMIC MODULUS, $ E^* $]				
Layer Number	c1	c2	c3	c4	d1	d2	d3	d4
1	3.123	3.446	-0.125	0.554	3.123	3.446	-0.800	-0.554
2								
3								

Sigmoidal Relaxation Modulus Function:

$$\log E(t) = c_1 + \frac{c_2}{1 + e^{c_3 + c_4 \log(t_r)}}$$

—————> Convert $E(t)$ to $|E^*|$

—————< Convert $|E^*|$ to $E(t)$

Sigmoidal Dynamic Modulus Function:			
$\log E^* = d_1 + \frac{d_2}{1 + e^{d_3 + d_4 \log(f)}}$			

PRONY SERIES COEFFICIENTS

Prony Coefficients				Sum of Squared Errors (log psi)		
Index No.	Ei			Layer1	Layer2	Layer3
	Layer1	Layer2	Layer3			
0	1.6E+03	0.0E+00	0.0E+00	0.0E+00	2.81E-02	0.00E+00
1	1.5E+02	0.0E+00	0.0E+00	1.0E+07	1.59E-01	0.00E+00
2	3.4E+02	0.0E+00	0.0E+00	1.0E+06		
3	6.7E+02	0.0E+00	0.0E+00	1.0E+05		
4	1.5E+03	0.0E+00	0.0E+00	1.0E+04		
5	3.9E+03	0.0E+00	0.0E+00	1.0E+03		
6	1.1E+04	0.0E+00	0.0E+00	1.0E+02		
7	3.6E+04	0.0E+00	0.0E+00	1.0E+01		
8	1.1E+05	0.0E+00	0.0E+00	1.0E+00		
9	2.9E+05	0.0E+00	0.0E+00	1.0E-01		
10	5.4E+05	0.0E+00	0.0E+00	1.0E-02		
11	6.9E+05	0.0E+00	0.0E+00	1.0E-03		
12	6.5E+05	0.0E+00	0.0E+00	1.0E-04		
13	4.9E+05	0.0E+00	0.0E+00	1.0E-05		
14	3.6E+05	0.0E+00	0.0E+00	1.0E-06		

Prony Series Resilient Modulus Function:

$$E(t) = E_0 + \sum_{i=1}^M E_i \cdot e^{-\frac{t}{\rho_i}}$$

Prony Series Dynamic Modulus Function:

$$|E^*(\omega)| = \sqrt{\{E'(\omega)\}^2 + \{E''(\omega)\}^2}$$

$$E'(\omega) = E_0 + \sum_{i=1}^M \frac{\varpi^2 \rho_i^2 E_i}{\varpi^2 \rho_i^2 + 1} \quad E''(\omega) = \sum_{i=1}^M \frac{\varpi \rho_i E_i}{\varpi^2 \rho_i^2 + 1}$$

Figure 39. Dynamic_Modulus_Calc Tab in ViscoWave Interface

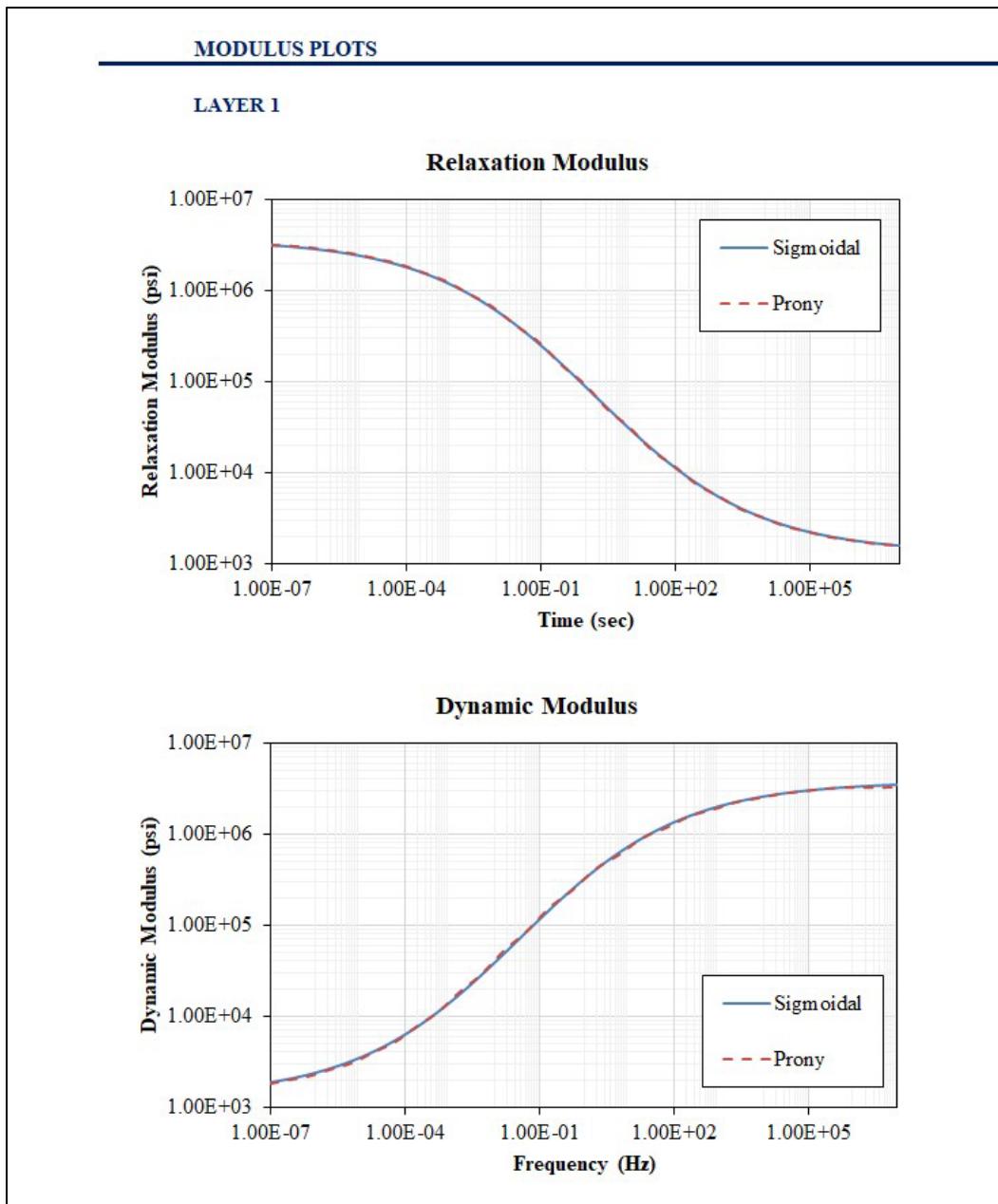


Figure 40. Relaxation and Dynamic Modulus Plots

BACKCALCULATION USING BATCH MODE

Batch mode backcalculation allows you to backcalculate multiple FWD drops. You can run either the **Constrained** or the **Unconstrained** backcalculation by clicking on the appropriate button in the main menu (Figure 25).

When you click on one of the batch options for backcalculation, you will be prompted to select the CSV files (produced from pre-processing) as shown in Figure 41. You may choose multiple files to be imported into ViscoWave and backcalculation conducted.

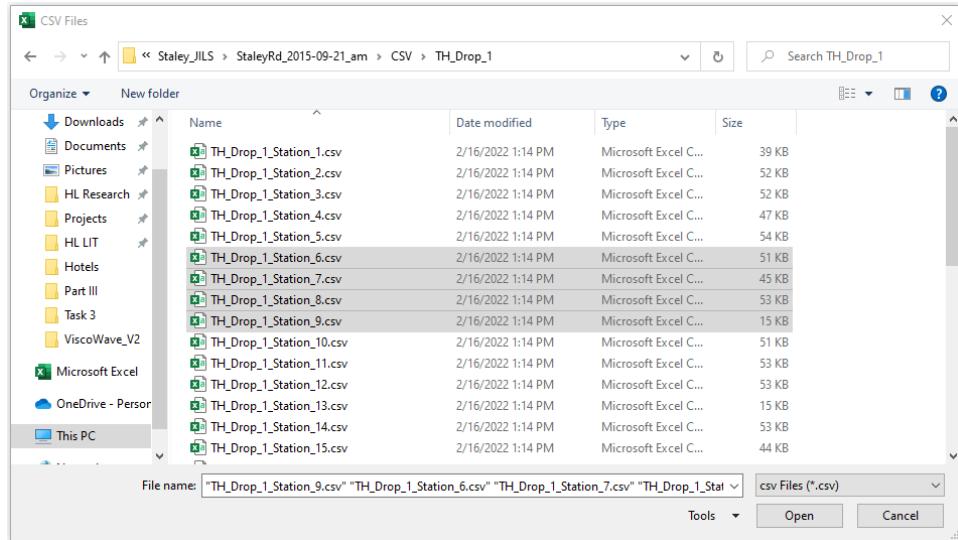


Figure 41. Selecting Multiple CSV Files for Batch Mode Backcalculation

Once the batch mode analysis is completed, the backcalculated ViscoWave files are saved in the same folder and with the same name as the CSV file (but will have *.xlsm extension), as shown in Figure 42.

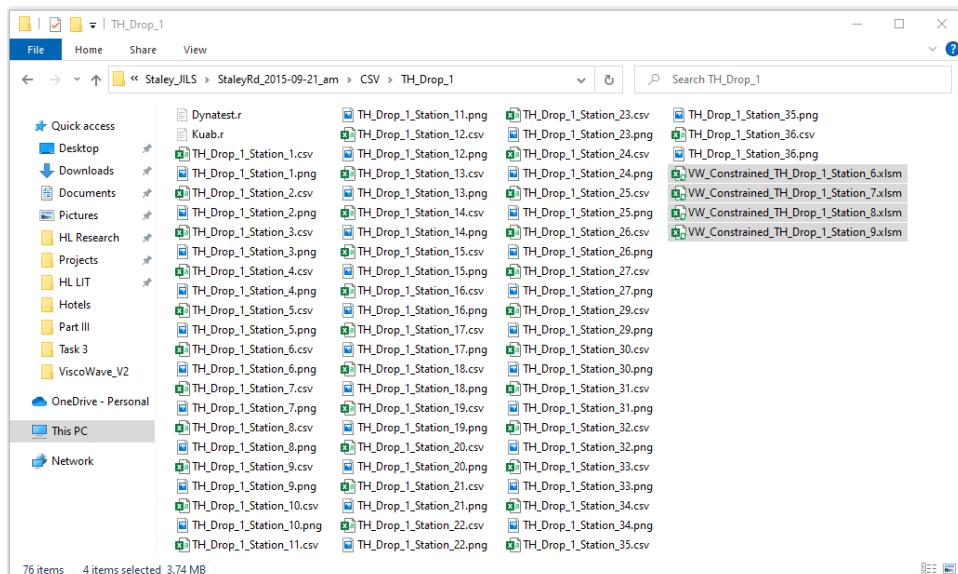


Figure 42. Files Created from Batch Mode Backcalculation

Important notes related to batch mode backcalculation.

- When you run batch mode, the **Seed values** stored in the "Default Inputs" worksheet are used as seed values for backcalculating all CSV files selected. Because Seed values are very important, **it is strongly recommended that you run manual backcalculation on at least one of the CSV files and export the satisfactory results to the "Default Inputs"** (i.e., Step 6 of Manual Backcalculation) before running batch mode backcalculation.

2. Similarly, in case of constrained backcalculation, the Constraints stored in the "Default Inputs" become active for all CSVs selected for batch mode processing.
3. The batch mode processing **automatically converts the Relaxation Modulus coefficients to Dynamic Modulus coefficients**.
4. As a reminder, the backcalculated ViscoWave file (.xlsm file) is saved automatically with the same name as the CSV file.

POST-PROCESSING FOR BATCH MODE BACKCALCULATION

Post-processing option gathers the backcalculated modulus from the individual ViscoWave (*.xlsm) files produced from batch processing, and calculates the average modulus for each layer. In addition, for the asphalt concrete (i.e., viscoelastic layer), the inputs needed for Mechanistic-Empirical (ME) design are calculated.

To run post-processing, click on the "**Generate Backcalculation Summary**" button and select the *.xlsm files produced from batch processing. The summary file is named "ViscoWave_Backcalculation_Summary.xlsm" and saved in the same folder as the individual ViscoWave files as shown in Figure 43.

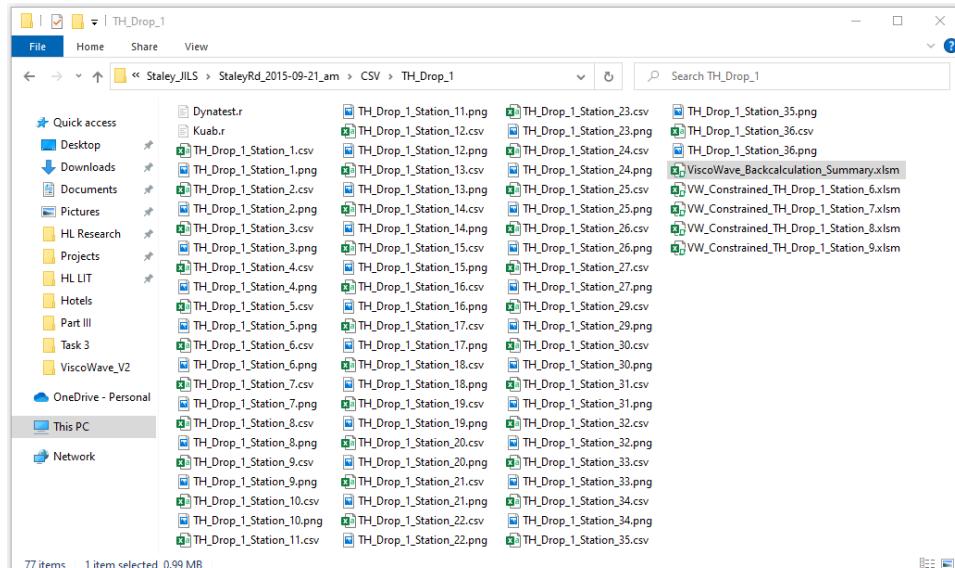


Figure 43. Summary File Created from Post-Processing

The summary file includes 2 additional worksheets, described as the following.

- **DM_Summary worksheet** (Figures 44 and 45) which includes the dynamic modulus from the individual backcalculation as well as the overall average dynamic modulus. The worksheet also provides the dynamic modulus values that can be used for ME design.

- If you need to change the reference temperature and/or the shift factor coefficients of the AC material, make sure you click the “Re-Calculate Coefficients” button again to re-generate the Dynamic Modulus coefficients (d_1 through d_4).
- **ElasticLayer_Summary worksheet** (Figures 46 and 47) which includes the elastic modulus from the individual backcalculation as well as the average modulus values for each elastic layer included in the pavement structure.



ViscoWave

Version 2.0
Developed and Implemented by Hyung S. Lee, Ph.D., P.E.

TIME-TEMPERATURE SHIFT (TTS) FACTOR INPUTS

Shift Factor Equation: $\log(a_T) = aT^2 + bT + c$ Also, since, $\log(a_T) = 0$ @ Reference Temp., $T=T_R$: $c = -aT_R^2 - bT_R$

Reference Temperature: 70.0 °F (21.1 °C)

Shift Factor Coefs: a: 0.0007 b: -0.1592 c: 3.0489

Note 1: The reference temperature only needs to be specified in deg. F. (It is automatically converted to deg. C)
Note 2: The temperature in the above TTS equation is in deg. C. (I.e., the TTS coefficients should be specified for T in deg. C.)
Note 3: The user only needs to specify a and b coefficients. The c coefficient is calculated automatically based on the above Eq.
Note 4: Default values for a and b are from Sakaefar, M.S., "Development of New Dynamic Modulus Predictive Models for Hot Mix Asphalt Mixtures", Ph.D. Dissertation, North Carolina State University, Raleigh, NC. 2011.

AVERAGE DYNAMIC MODULUS COEFFICIENTS

Sigmoidal Coefficients [DYNAMIC MODULUS, (E*)]							
Layer #	Description	d1	d2	d3	d4	Temp, F	Temp, C
1	@FWD Test Temperature	2.844	3.739	-0.657	-0.538	72.1	22.3
1	@Reference Temperature	2.844	3.739	-0.736	-0.538	70.0	21.1

Re-Calculate Coefficients

MECHANISTIC-EMPIRICAL (ME) DESIGN INPUTS

Dynamic Modulus, [E*], psi						
Temperature		Frequency, Hz				
°C	°F	0.1	0.5	1	5	10
-10.0	14	2.3E+06	2.6E+06	2.8E+06	3.1E+06	3.2E+06
4.4	40	7.6E+05	1.2E+06	1.4E+06	1.8E+06	2.0E+06
21.1	70	7.9E+04	1.7E+05	2.4E+05	4.5E+05	5.8E+05
37.8	100	9.0E+03	1.9E+04	2.6E+04	5.8E+04	8.2E+04
54.4	130	2.6E+03	4.2E+03	5.4E+03	1.0E+04	1.4E+04

Reduced Frequency, Hz						
Temperature		Frequency, Hz				
°C	°F	0.1	0.5	1	5	10
-10.0	14	3.7E+00	4.4E+00	4.7E+00	5.4E+00	5.7E+00
4.4	40	1.4E+00	2.1E+00	2.4E+00	3.1E+00	3.4E+00
21.1	70	-1.0E+00	-3.0E-01	0.0E+00	7.0E-01	1.0E+00
37.8	100	-3.0E+00	-2.3E+00	-2.0E+00	-1.3E+00	-9.7E-01
54.4	130	-4.5E+00	-3.8E+00	-3.5E+00	-2.8E+00	-2.5E+00

Shift Factor, log(a _T)		
Temperature	log(a _T)	
°C	°F	
-10.0	14	4.7E+00
4.4	40	2.4E+00
21.1	70	0.0E+00
37.8	100	-2.0E+00
54.4	130	-3.5E+00

Figure 44. Dynamic Modulus Summary for Viscoelastic Layers

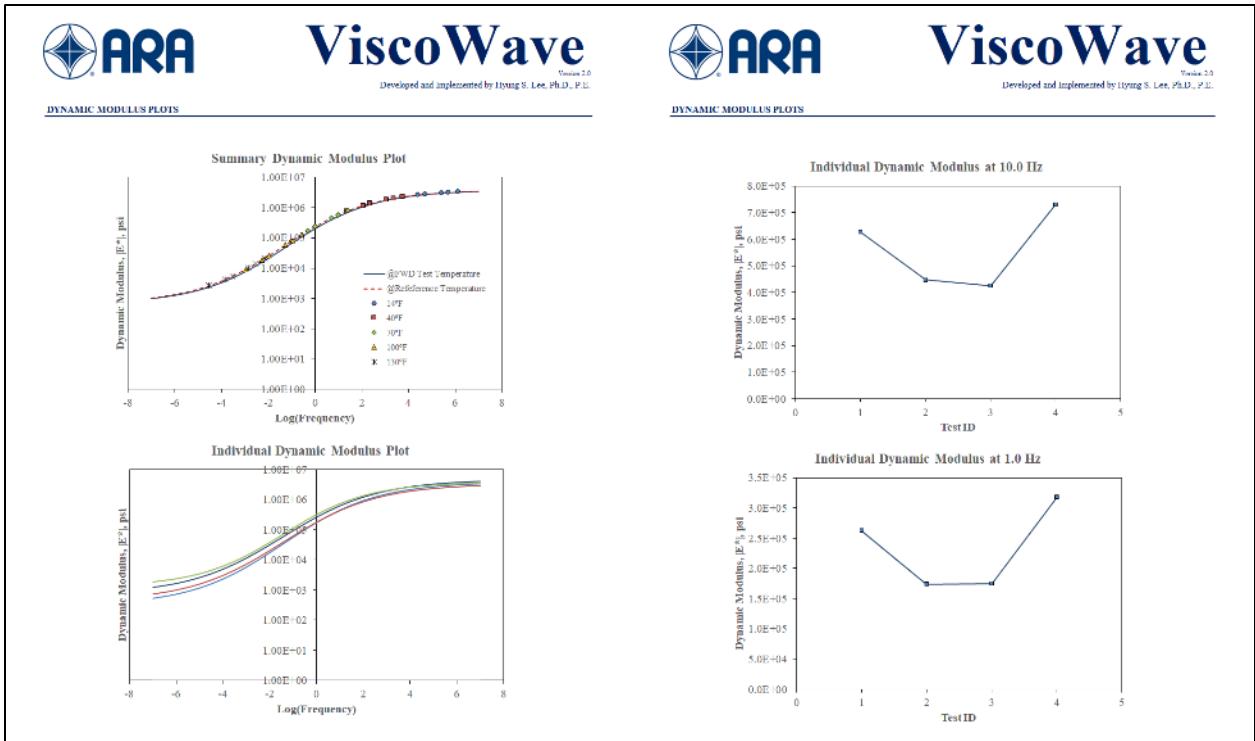


Figure 45. Plots of Backcalculated Dynamic Modulus



ViscoWave

Version 2.0

Developed and Implemented by Hyung S. Lee, Ph.D., P.E.

ELASTIC MODULUS SUMMARY

Modulus Summary

Layer Number	Mean	Std.Dev.	Min.	25%-tile	75%-tile	Max
1	0	0	0	0	0	0
2	9634	3623	5000	7459	11443	15000
3	10265	522	9565	9891	10659	10924
4						
5						
6						

Note 1: If a layer was modeled as viscoelastic, the results of that layer will not be shown in this worksheet.

INDIVIDUAL BACKCALCULATION RESULTS

Note 1: Remove outliers by deleting Cell Contents. **Do NOT Delete or Remove Excel Rows!**

Backcalculated Sigmoidal Coefficients

Test ID	File Name	1	2	3	4	5	6
1	VW_Constrained TH_Drop_1_Station_6.xls	0	5000	9565			
2	VW_Constrained TH_Drop_1_Station_7.xls	0	8278	10924			
3	VW_Constrained TH_Drop_1_Station_8.xls	0	10257	10570			
4	VW_Constrained TH_Drop_1_Station_9.xls	0	15000	10000			

Figure 46. Elastic Modulus Summary



ViscoWave

Version 2.0

Developed and Implemented by Hyung S. Lee, Ph.D., P.E.

DYNAMIC MODULUS PLOTS

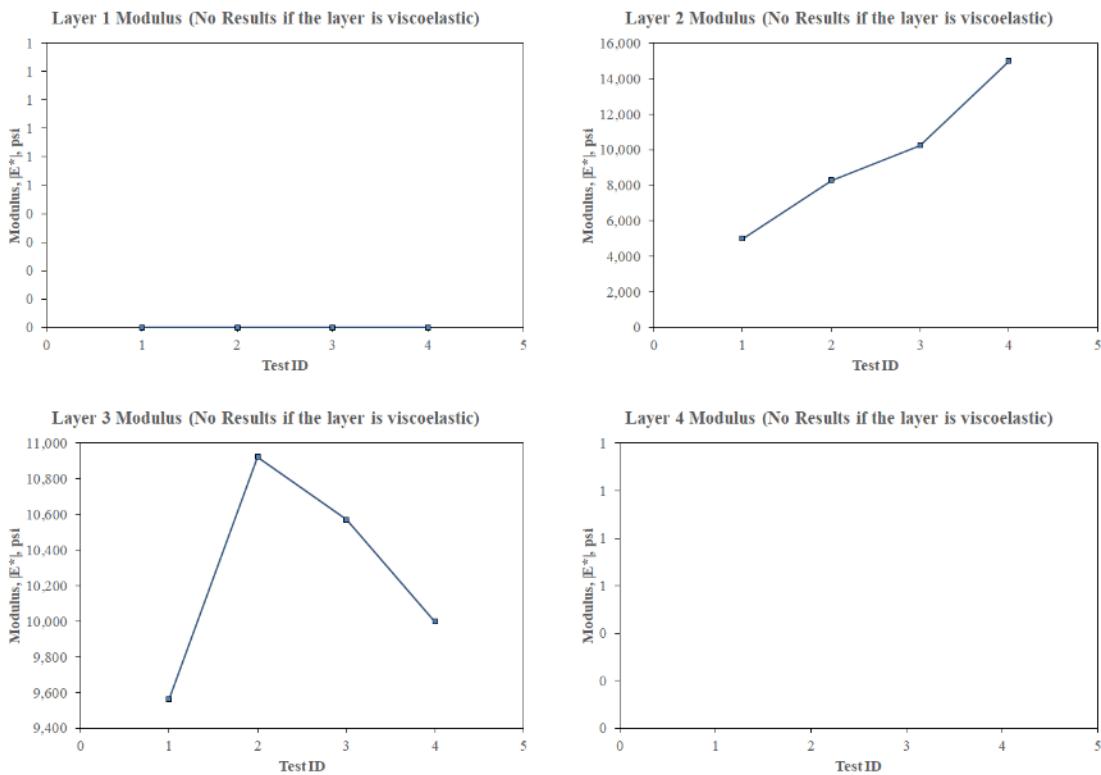


Figure 47. Plots of Backcalculated Elastic Modulus

REFERENCES

- Al-Khoury, R., Scarpas, A., Kasbergen, C., and Blaauwendraad, J. 2001a. Spectral Element Technique for Efficient Parameter Identification of Layered Media. I. Forward Calculation. International Journal of Solids and Structures. 38: 1605-1623.
- Al-Khoury, Kasbergen, C., R., Scarpas, A., and Blaauwendraad, J. 2001b. Spectral Element Technique for Efficient Parameter Identification of Layered Media Part II. Inverse Calculation. International Journal of Solids and Structures. 38: 8753-8772.
- Al-Khoury, R., Scarpas, A., Kasbergen, C., and Blaauwendraad, J. 2002. Spectral Element Technique for Efficient Parameter Identification of Layered Media Part III. Viscoelastic Aspects. International Journal of Solids and Structures. 39: 2189-2201.
- Applied Research Associates (ARA). 2004. Guide for Mechanistic-Empirical Design of New and Rehabilitated Pavement Structures. NCHRP Project 1-37A, Final Report, National Cooperative Highway Research Program, Transportation Research Board, National Research Council, Washington, DC.
- Al-Qadi, I., Xie, L., and Elseifi, M. A. 2008. Frequency Determination from Vehicular Loading Time Pulse to Predict Appropriate Complex Modulus in MEPDG, Journal of the Asphalt Paving Technologists, Vol. 77, pp. 739-772.
- Beale, M.H., Hagan, M.T., and Demuth, H.B. 2011. Neural Network ToolboxTM User's Guide, MathWorks, Inc., Natick, MA.
- Chatti, K., and Yun, K. K. 1996. SAPSI-M: Computer Program for Analyzing Asphalt Concrete Pavements Under Moving Arbitrary Loads. Transportation Research Record, 1539: 88-95.
- Chatti, K., Ji, Y., and Harichandran, R. 2004. Dynamic Time Domain Backcalculation of Layer Moduli, Damping and Thicknesses in Flexible Pavements. Transportation Research Record, 1869: 106-116.
- Christensen, R. M. 2003. Theory of Viscoelasticity. 2nd Ed. Dover Publications, Inc. NY.
- Findley, W. N., Lai, J. S., and Onaran, K. 1976. Creep and relaxation of nonlinear viscoelastic materials. Dover Publications, Inc. NY.
- Hadidi, R., and Gucunski, N. 2007. A Probabilistic Approach to Falling Weight Deflectometer Backcalculation. Transportation Research Board Annual Meeting CD-ROM, 07-3187, Transportation Research Board, Washington D. C.
- Ji, Y. 2005. Frequency and time domain backcalculation of flexible pavement layer parameters. Ph. D. Dissertation. Michigan State University.
- Kim, Y. R. 2009. Modeling of Asphalt Concrete. ASCE Press. McGraw Hill.

Kim, Y.R., Underwood, B., Sakhaei Far, M. Jackson, N., and Puccinelli, J. 2011. LTPP Computed Parameter: Dynamic Modulus. FHWA-HRT-10-035. Turner Fairbank Highway Research Center, Federal Highway Administration, McLean, VA.

Kutay, E., Chatti, K., and Lei, L. 2011. Backcalculation of Dynamic Modulus ($|E^*|$) Mastercurve from FWD Surface Deflections. Transportation Research Board Annual Meeting DVD, 11-3434, Transportation Research Board, Washington D. C.

Lee, H. S. 2011, ViscoWave – A New Forward Solution for Dynamic Backcalculation. A Presentation Made at the 20th Annual FWD Users Group Meeting.
<http://pms.nevadadot.com/2011.asp>. Last Accessed July 2016.

Lee, H. S. 2013. Development of a New Solution for Viscoelastic Wave Propagation of Pavement Structures and Its Use in Dynamic Backcalculation. Ph. D. Dissertation. Michigan State University.

Lee, H. S. 2014. ViscoWave – A New Solution for Viscoelastic Wave Propagation of Layered Structures Subjected to an Impact Load. International Journal of Pavement Engineering, 15(6): 542-557.

Lee, H. S. 2016. Use of a Dynamic, Viscoelastic Finite Layer Model for Simulation of Asphalt Pavement Response, Presented at the 95th Transportation Research Board, AFD80 Committee Meeting, Washington, D.C.

Lee, H.S., Ayyala, D., and Von Quintus, H. 2017. Dynamic Backcalculation of Viscoelastic Asphalt Properties and Master Curve Construction. Transportation Research Record, No. 2641. DOI 10.3141/2641-05.

Matsui, K., Maina, J. W., Hachiya, Y., and Ozawa, Y. 2011. Wave Propagation Analysis for Flexible Pavements and Its Application to Backcalculation Analysis. Transportation Research Board Annual Meeting DVD, 11-2971, Transportation Research Board, Washington D. C.

Sakhaifar, M.S. 2011. Development of New Dynamic Modulus Predictive Models for Hot Mix Asphalt Mixtures. Ph.D. Dissertation, North Carolina State University, Raleigh, NC.

Siddharthan, R.V., Yao, J., and Sebaaly, P.E., Pavement Strain from Moving Dynamic 3-D Load Distribution, Journal of Transportation Engrg., ASCE, Vol. 124(6), Nov./Dec. 1998, pp. 557-566.

Siddharthan, R.V., Krishnamenon, N., and Sebaaly, P.E., Finite-Layer Approach to Pavement Response Evaluation, Transportation Research Record No. 1709, TRB, 2000, pp. 43-49.

Siddharthan, R., Sebaaly, P.E., El-Desouky, M., Strand, D., and Huft, D. "Heavy Off-road Vehicle Tire-Pavement Interactions and Response," Journal of Transportation Engineering, ASCE, Vol. 131(3), March/April 2005, pp. 239-247.