Software Requirements Specification for Two and Three Dimensional Dynamic Model of Soil-Water-Structure Interaction

Prepared by:
Brandon Karchewski (karcheba@mcmaster.ca)
Ph.D. Candidate
Department of Civil Engineering

Prepared for:
Dr. Spencer Smith
CES 741 - Development of Scientific Computing Software
Department of Computational Engineering and Science

McMaster University
Hamilton, Ontario, Canada

Ver. DynSWS-SRS-1.0
April 4, 2012

Table of Contents

Table of Symbols

Table 0.1: Document list prefixes

Symbol	Description
PS	Physical System Description
G	Goal Statement
A	Assumption
TM	Theoretical Model
IM	Instanced Model
SB	System Behaviour
NFR	Non-functional Requirement

Table 0.2: Superscripts and subscripts

Symbol	Description
a^f, a_f	Indicates that symbol a refers to the fluid phase
a^s, a_s	Indicates that symbol a refers to the solid phase
a_{MIN}	Indicates a system imposed constraint on the minimum value of variable a
a_{MAX}	Indicates a system imposed constraint on the maximum value of variable a

Table 0.3: Physical units and mathematical definitions

Symbol	Description
M	Symbolic unit for mass (*e.g.* kilograms)
L	Symbolic unit for distance (*e.g.* metres)
T	Symbolic unit for time (*e.g.* seconds)
t	The temporal dimension
x, y, z	The three spatial dimensions in the Cartesian coordinate system
$\frac{\partial a}{\partial s}$	The derivative of function a with respect to variable s
a_i	A vector with either two components ($i \in \{x, y\}$) or three components ($i \in \{x, y, z\}$)
a_{ij}	A second-order tensor with either four components ($i, j \in \{x, y\}$) or nine components($i, j \in \{x, y, z\}$)
$\dot{a}$	The first derivative of a with respect to time
$\ddot{a}$	The second derivative of a with respect to time
$a_{,i}$	The first derivative of a with respect to spatial coordinate i
$a_{,ij}$	The derivative of a with respect to spatial coordinates i and j in that order
a_{ii}	A repeated index implies summation; *e.g.* for $i \in \{x, y, z\}$, $a_{ii} = a_{xx} + a_{yy} + a_{zz}$
δ_{ij}	The Kronecker delta, which has the value one for $i = j$ and zero otherwise
∇	The del operator as defined for the coordinate system in use (*e.g.* for Cartesian coordinates, $\left\langle \frac{\partial}{\partial x}, \frac{\partial}{\partial y}, \frac{\partial}{\partial z} \right\rangle$)

Table 0.4: Domains and boundaries

Symbol	Description
$\mathbb{R}^k$	Symbol for k-dimensional space; typically, $k \in \{2, 3\}$
$\boldsymbol{\Omega}$	Total domain of soil-water-structure interaction problem
$\boldsymbol{\Omega_{st}}$	Subdomain of structural elements; single phase solid
$\boldsymbol{\Omega_{w}}$	Subdomain of water; single phase fluid
$\boldsymbol{\Omega_{sw}}$	Subdomain of soil; two phases: solid particles and pore water
$\boldsymbol{S}$	Total system boundary; excludes boundaries between subdomains
$\boldsymbol{S_{st}}$	Boundary of structural subdomain
$\boldsymbol{S_{w}}$	Boundary of water subdomain
$\boldsymbol{S_{sw}}$	Boundary of soil subdomain

Table 0.5: Material properties

Symbol	Description
E	Elastic modulus of a solid
n	Porosity; the volume fraction of a two phase material (*e.g.* soil-water) taken up by the pore fluid
n_{DENSE}	The porosity at which the behaviour of the solid particle phase transitions from Newtonian to non-Newtonian behaviour
β	Nonlinear drag coefficient; related to momentum transfer between solid and fluid phases
μ	Viscosity of a Newtonian fluid
ν	Poisson's ratio of a solid
ρ	Density; mass per unit volume

Table 0.6: Field variables

Symbol	Description
$u_i, \dot{u}_i, \ddot{u}_i$	Displacement, velocity, and acceleration vectors
σ_{ij}	Stress tensor; force per unit area
ε_{ij}	Strain tensor; normalized measure of deformation
$\dot{\varepsilon}_{ij}$	Strain rate tensor; normalized rate of deformation per unit of time
f_i	Vector of values that yield applied forces when multiplied by mass
p	Pressure; the component of normal stress that is the same in all directions

Acronyms and Abbreviations

2-D/3-D: Two-dimensional/three-dimensional; refers to the dimension of the coordinate system used to solve the problem.

CFD: Computational Fluid Dynamics; a broad term encompassing numerical techniques for solving fluid dynamics problems.

DynSWS: Dynamic model of Soil-Water-Structure interaction; the software product described herein.

FEA/FEM: Finite Element Analysis/Finite Element Method; a numerical technique for solving systems of PDEs.

PDE: Partial differential equation. Models of physical phenomena are typically stated mathematically as systems of this type of equation that must be integrated to obtain the solution. Initial and/or boundary conditions are also required to solve the problem.

SRS: Software requirements specification; the name attributed to this type of document.

Quick Reference Tables

Table 0.7: Index of Physical System Description

Item	Description	Page
PS1	Domain description	11
PS2	Boundary description	11
PS3	Body forces	11
PS4	Initial conditions	12
PS5	Boundary conditions	12

Table 0.8: Index of Goal Statements

Item	Description	Page
G1	Compute displacement, velocity, and acceleration field	12
G2	Compute stress and strain field	12

Table 0.9: Index of Assumptions

Item	Description	Page
A1	Isothermal	13
A2	Low velocities (non-relativistic)	13
A3	No material sources or sinks	13
A4	Continuum representation	13
A5	Maximum of two material phases	13
A6	Homogeneous materials	14
A7	Isotropic materials	14
A8	Impermeable structure	14
A9	Small strains	14
A10	Linear elastic structure	14
A11	Water is incompressible	14
A12	Water is a Newtonian fluid	14
A13	Laminar flow	15
A14	Small strain rates	15
A15	Soil grains are incompressible	15
A16	For $n < n_{DENSE}$, soil behaviour is non-Newtonian	15
A17	Non-Newtonian behaviour of soil is linear elastic	15
A18	For $n \geq n_{DENSE}$, soil behaviour is Newtonian	15
A19	Cartesian coordinates	15
A20	2-D model is plane strain	16

Table 0.10: Index of Theoretical Models

Table 0.11: Index of Instanced Models

Table 0.12: Index of System Behaviour

Table 0.13: Index of Non-Functional Requirements

1 Introduction

This section gives a general outline of the SRS for DynSWS, where DynSWS is the software product discussed herein. Section 1.1 provides the reason for creating the SRS as well as the audience that it is directed toward. Section 1.2 gives a high-level view of what DynSWS is intended to do as well as explicit statements of what it will not do. Section 1.3 explains the structure of the document as compared to a base reference for SRS document structure.

1.1 Purpose of the Document

This document outlines the requirements for a scientific computing model of the interaction between a structure, its foundation (composed primarily of soil and/or rock), and a body of water under dynamic loading. The most pertinent example is a dam-reservoir system subjected to seismic activity. It is important to note that this document focuses on the nature of the model and its intended functions and characteristics, but leaves the details of how to accomplish these goals to a later stage in the development process.

Development of the model is a component of the author's doctoral thesis, so the primary audience is the author himself as well as his supervisory committee. The project may involve interaction with an industry partner that develops and maintains a geotechnical software product. This means that compatibility with existing pre- and post-processing software as well as the needs of the industry partner will be important considerations.

1.2 Scope of the Software Product

The main objective of the computational model is to capture the behaviour of a physical system involving a structure, its foundation, and an adjacent body of water when subjected to prescribed ground motion. To this end, the model must be able to solve for the variation of field variables including displacement, velocity, acceleration, stress, and strain over time given the geometry of the system, material properties, mathematical models describing the system, initial conditions, and boundary conditions. The primary benefit of accurately modelling such behaviour is the improved understanding of the interaction between the different physical components, which will lead to recommendations for better practices in the design of such structures with regard to dynamic loading. The development of DynSWS will involve verification and validation against real-world case examples as well as physical laboratory testing (see Section 3.3.4).

Solutions of such systems typically involve numerical approximations of the mathematical models over a discretized version of the problem domain. DynSWS will not be responsible for generating this discretized domain (or "mesh") and will rely on an external source for this functionality, possibly provided by the industry partner. Analysing the results of the computer model will also involve graphical representation of the output through items such as line plots and contour plots. DynSWS will not be responsible for generating graphical output and will again, rely on an external software product to provide this functionality.

1.3 Organization of the Document

The organization of this SRS closely follows the format set out in [1], which contains an SRS template specifically tailored to the needs of scientific computing software. Note that the Terminology Definition (Section 3.1.2) has been further divided into Software Engineering Terminology (Section 3.1.2.2) and Domain Specific Terminology (Section 3.1.2.1) so that readers already familiar with one or the other set of definitions may browse for unfamiliar terms more easily.

2 General System Description

This section provides general information on DynSWS and its environment. Section 2.1 outlines the environment of DynSWS and how it will interface with other software products necessary for its operation. Section 2.2 describes the intended users of the product. Section 2.3 lists any items that may limit implementation options for DynSWS.

2.1 System Context

Figure 2.1 shows the context of the DynSWS software product. It is clear in the diagram that the software product will follow the traditional work-flow of scientific computing software, which is: get input, perform calculations, produce output. Circles highlight human interaction with the software product and its environment. Rectangles represent software products including the system itself. Note that there will be no direct human interaction with the DynSWS system as it will rely on external software products to perform pre- and post-processing of the data.

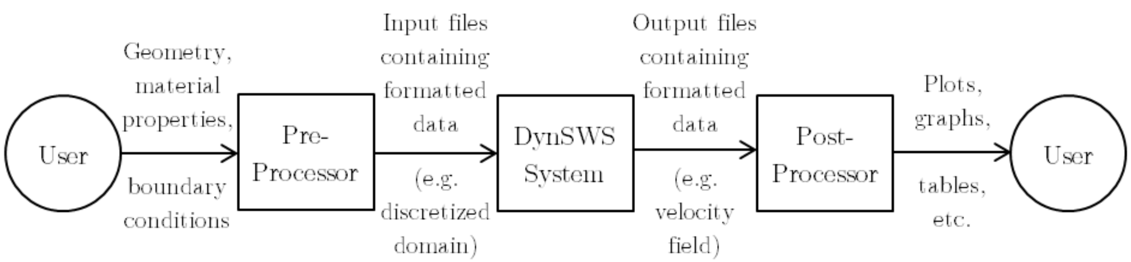

Figure 2.1: System context diagram for DynSWS

Before describing theoretical and practical details of the DynSWS system, it is worthwhile to delineate the responsibilities of each component as shown in Figure 2.1. The responsibilities are as follows:

1. User Responsibilities
   - Understand the nature of the problem, the underlying theory of the model, and the implications of the numerical solution technique
   - Prepare a set of input data including problem geometry, material properties, and boundary conditions
   - Ensure that the input information is accurate and makes use of a consistent unit system
   - Interpret the output information in the context of the theoretical and numerical model
   - Perform the necessary parametric and sensitivity analysis to understand the behaviour of the system

2. Pre-Processor Responsibilities
    - Facilitate input of problem data
    - Take reasonable measures to prevent data entry errors and to ensure units are consistent
    - Perform any necessary discretization of the problem domain
    - Generate formatted input files for the DynSWS system
    - Ensure that a complete data set has been entered and that a complete set of input files has been generated
    - Perform auxiliary tasks such as saving and loading models.
3. DynSWS Responsibilities
    - Read data from formatted input files
    - Ensure that all necessary input files and data are present and in correct format
    - Verify that input data does not violate any constraints
    - Perform analysis to determine the desired outputs
    - Ensure that output from the model is sensible. That is, the output is either a converged solution or an error message. Note that this says nothing of whether the converged solution is the “true” solution as the true solution is typically not known, which is the reason for using a numerical model.
    - Generate formatted output files containing results (including error messages, if any)
4. Post-Processor Responsibilities
    - Read data from formatted output files
    - Ensure that all necessary output files and data are present and in correct format
    - Allow user to interact and select desired output plots and tables.
    - Generate and display selected output data
    - Perform auxiliary tasks such as saving plots and printing.

2.2 Stakeholder Characteristics

The primary user of the DynSWS system during the development stage will be the author in carrying out thesis-related research. As such, this user will have an intimate knowledge of the underlying theory as well as the implementation details of the DynSWS system and (at minimum) knowledge of the interface with the pre- and post-processor components in the form of the input and output file formats. In terms of background knowledge, the author has completed an undergraduate degree in civil engineering and related coursework at the graduate level in civil engineering and computational mechanics.

Another important group of users will be members of the author's supervisory committee. These users will be familiar with the underlying theoretical models. Individual members of this group will have detailed knowledge of some components of the system, but may not necessarily have detailed knowledge of the entire implementation at all times. Members of this group of stakeholders are university professors with expert knowledge in fields including geotechnical engineering, structural engineering, and software engineering.

As mentioned in Section 1, the development of DynSWS may involve interaction with an industry partner. This adds two additional groups of stakeholders to the software product: software developers at the partner company and potential users of the software product as implemented in the commercial code. The former will certainly have knowledge of computer programming and software engineering practices; this group may also have specialized knowledge of geotechnical engineering and engineering mechanics as it relates to the software product that they maintain. The latter group is likely to possess a post-secondary education in engineering at either the college or university level. While members of these groups may not necessarily possess detailed knowledge of the underlying mathematical or numerical models, they should have some background in dynamics and numerical solution techniques for systems of partial differential equations, which is typically obtained through an undergraduate education in engineering.

2.3 System Constraints

The main constraints for DynSWS will derive from interaction with other software products. The industry partner develops their software product for both Windows and Macintosh users so DynSWS must be able to run on both systems. Also see Section 4.4 for potential constraints that are not well defined at this point.

3 Specific System Description

This section provides the specifics of the DynSWS system requirements. Section 3.1 provides an overview of the type of problem that the product is intended to solve, an introduction to the general concepts involved in the solution, and goals detailing what the solution should look like. Section 3.2 examines the details of what characteristics the solution should have including the models employed, important assumptions of the models, and other specifics related to implementing the solution. Section 3.3 describes other characteristics that the software product should have, which, while not directly related to the actual solution of the problem, are of equal importance in ensuring that the software product is successful and long-lived.

3.1 Problem Description

This section gives general background information on the type of problems that DynSWS is intended to solve. Section 3.1.1 outlines the general background of the class of problem and some reasons why the solution is important. Section 3.1.2 defines the key terms that are necessary to understand the theoretical models employed in the solution and the software engineering design of the product. Section 3.1.3 introduces a schematic representation of the problem and describes the key components of the physical system that require solution. Section 3.1.4 lists the most important functional goals for the solution of the problem.

3.1.1 Background Overview

DynSWS is intended to solve problems involving the dynamic excitation of physical systems involving interaction between a structure, its soil or rock foundation, and a body of water. Figures 3.1 and 3.2 show photos of such systems, both of which are prone to risk of seismic excitation as they are in coastal regions near tectonic plate boundaries.

Figure 3.1 depicts a large structure with a foundation composed of soil and/or rock that is located in the path of flowing water resulting in the accumulation of water upstream of the structure. Figure 3.2 is an offshore structure surrounded by ocean water and founded on the seabed, also composed of soil and/or rock. The former maintains the presence of an all-important resource for human settlements (fresh water) and the latter facilitates the extraction of a resource important to modern industrial economies (hydrocarbons). The significance of the resources that such systems provide and the consequences of their failure mean that a proper understanding of their behaviour under dynamic excitation is critical to ensuring safe engineering designs.

3.1.2 Terminology Definition

This section defines a number of key terms necessary to understand the problem formulation and the description of the solution. It is divided into two parts: Domain Specific Terminology (Section 3.1.2.1) and Software Engineering Terminology (Section 3.1.2.2). The reason for the division is so that readers familiar with one subject area do not need to sort through a combined list for terms that they are unfamiliar with.

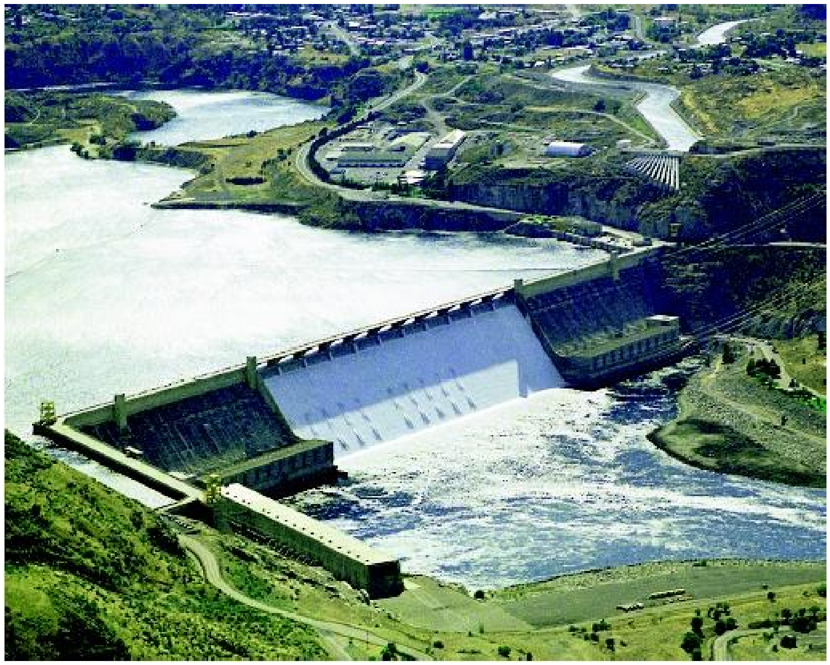

Figure 3.1: The Grand Coulee dam-reservoir system in Okanagan County, Washington, USA (*Image Source:* [2])

Figure 3.2: The Hibernia oil rig off the coast of Newfoundland, Canada (*Image Source:* [3])

3.1.2.1 Domain Specific Terminology

Acceleration: The rate at which the velocity of a particle is changing and its associated direction at a given point in time; the second derivative of displacement with respect to time.

Dam: A type of retaining structure designed to hold back water flowing in a river; may be constructed of various materials such as soil, timber, or concrete.

Displacement: The distance of a particle from its initial position at a given point in time; perhaps resolved into its 2-D or 3-D components.

Field variable: A quantity that varies over a given domain; in the true solution this variation is continuous, while in an approximate solution its value may only be known at discrete points.

Foundation: A structural component that transfers loads from the human-made portion of a structure to the surrounding natural materials (typically soil and/or rock).

Material point: From continuum mechanics, the coordinates of the problem domain in the initial configuration.

Reservoir: A body of water that forms behind a dam due to the flow restriction imposed by the structure.

Retaining structure: A structure that holds back soil, rock, and/or water.

Rock: Natural material composed of minerals and formed mainly through the cooling of molten material from inside the Earth or the sedimentation and cementation of mineral fragments and/or chemical precipitates.

Seismicity: Ground motion induced by the propagation of waves due to earthquakes.

Soil: Natural material composed of minerals, rock fragments, and organic matter; sometimes referred to as unconsolidated rock owing to the fact that it is a collection of discrete fragments. The strength of such materials derives from internal friction, which is sensitive to the applied pressure, and cohesion, which is a constant derived from physical and chemical bonds between particles.

Strain: A normalized measure of material deformation on a given plane; typically unitless, but may carry units of length per length.

Stress: A measure of the force per unit area on a given plane commonly resolved into components perpendicular to the plane ("normal" stress) and parallel to the plane ("shear" stress).

Structural dynamics: The study of how structures behave when subjected to loading that varies over a relatively short period of time.

Velocity: The rate at which a particle, or material point, is moving and its associated direction at a given point in time; the first derivative of displacement with respect to time.

3.1.2.2 Software Engineering Terminology

Accuracy: A quality that quantifies how close data is to some standard for comparison such as a true solution or a benchmark problem. For example, the number of significant digits of π carried by a floating point representation in a computer.

Functional Goal/Requirement: A type of goal or requirement that defines the primary purpose of a product. For example, a functional goal for a desktop calculator is to perform basic arithmetic given the necessary inputs. Often this type of requirement is met by creating a module that performs the required function.

Goal: An abstract intention for the operation of a software product stated in plain language. A goal may be imprecise and ambiguous, but states a desired outcome in an intuitive sense.

Maintainability: The ease with which a software product can be extended or altered when design decisions (*e.g.* model scope or data structure) are modified.

Non-functional Goal/Requirement: A type of goal or requirement that, while important for the success of a product, is not its primary function. For example, a non-functional requirement of a desktop calculator might be that the user interface should be intuitive and easy to manipulate. This type of requirement is not typically met by a single module or function, but rather must permeate the entire system.

Performance: A quantitative measurement of the time that it takes a system to obtain a solution. Ideally, this should either specify both a specific time constraint and the hardware or specify the time constraint in a general form understood to exclude machine-dependent constants.

Portability: The ability of a software product to operate on various underlying software or hardware architectures. For example, whether a software product can run on both Microsoft Windows and Mac OS systems.

Requirement: A detailed statement of a desired property (function or non-functional) of a software product. A good requirement should be as specific as possible and, ideally, possible to prove.

Sensitivity: A system property that describes how much the solution changes given a small change in input. For example, for $x \in [0.001, 0.002]$ the function $f(x) = x^{-1}$ is much more sensitive than the function $f(x) = x$.

Security: The ability of a software product to resist user access that is undesirable. For example, a software product that performs encryption may contain an encryption key; uses should not be able to access this value since it would allow them to circumvent the encryption techniques.

Usability: A system property that describes how easy a system is to operate for someone with the pre-requisite knowledge. For example, a user familiar with statistical theory should be able to become proficient with a statistical software package in a relatively short period of time.

Validation: The process of ensuring that a system is successful in achieving the solution to a problem as compared to some standard such as an analytical solution or laboratory test results. For a scientific computing problem, this aspect asks the question, "Were the correct equations solved?" *Note:* For scientific computing software, it is often difficult to separate validation from verification since in many cases the reason for creating the software product is to obtain the solution; that is, analytical solutions and/or actual test data may not be available. Consequently, errors in the underlying theory are difficult to separate from simple mistakes in the implementation.

Verification: The process of ensuring that a system performs its operations as specified. That is, checking that the algorithm carries out the individual operations in the specified manner. For a scientific computing problem, this aspect asks the question, "Were the equations solved correctly?" *See note in Validation.*

3.1.3 Physical System Description

This section presents the physical representation of the general class of problems that DynSWS is intended to solve. Figure 3.3 shows a schematic of a soil-water-structure system. Note that while a 2-D system is depicted, DynSWS will solve such systems in both 2-D and 3-D.

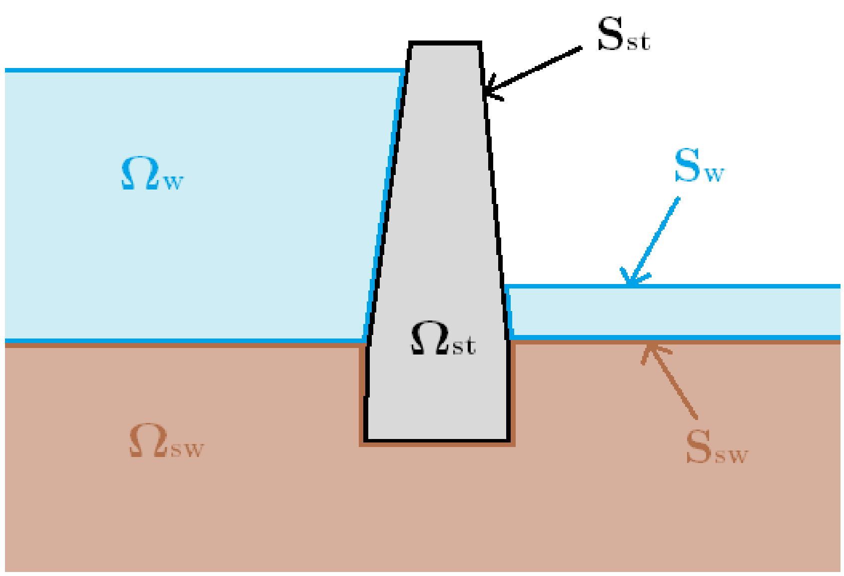

Figure 3.3: A schematic representation of a physical system involving soil-water-structure interaction

The important physical components of the system are as follows:

PS1. Problem Domain

PS1a. Reference Domain. The reference domain is $\mathbb{R}^k$ where $k \in \{2, 3\}$.

PS1b. Structural Subdomain The structural subdomain is represented by $\boldsymbol{\Omega_{st}}$, which is a subset of $\mathbb{R}^k$ of dimension k. This subdomain consists of single phase solids.

PS1c. Water Subdomain. The water subdomain is represented by $\boldsymbol{\Omega_w}$, which is a subset of $\mathbb{R}^k$ of dimension k. This subdomain consists of a single phase Newtonian fluid.

PS1d. Soil Subdomain. The soil subdomain is represented by $\boldsymbol{\Omega_{sw}}$, which is a subset of $\mathbb{R}^k$ of dimension k. This subdomain consists of two phases: solid particles and pore fluid.

PS1e. Total Domain. The total problem domain is $\boldsymbol{\Omega} = \boldsymbol{\Omega_{st}} \cup \boldsymbol{\Omega_w} \cup \boldsymbol{\Omega_{sw}}$. Note that $\boldsymbol{\Omega_{st}} \cap \boldsymbol{\Omega_w} = \boldsymbol{\Omega_{st}} \cap \boldsymbol{\Omega_{sw}} = \boldsymbol{\Omega_w} \cap \boldsymbol{\Omega_{sw}} = \emptyset$ since there should be no overlap between subdomains.

PS1f. Connectedness. Each of the subdomains $\boldsymbol{\Omega_{st}}$, $\boldsymbol{\Omega_w}$, and $\boldsymbol{\Omega_{sw}}$ may possibly be disconnected, but $\boldsymbol{\Omega}$ must be connected.

PS2. Problem Boundary

PS2a. Structural Boundary. The boundary of $\boldsymbol{\Omega_{st}}$ is given by $\boldsymbol{S_{st}}$, which is a subset of $\mathbb{R}^k$ of dimension $k-1$. That is, a set of line segments in $\mathbb{R}^2$ or a set of surfaces in $\mathbb{R}^3$.

PS2b. Water Boundary. The boundary of $\boldsymbol{\Omega_w}$ is given by $\boldsymbol{S_w}$, which is a subset of $\mathbb{R}^k$ of dimension $k-1$.

PS2c. Soil Boundary. The boundary of $\boldsymbol{\Omega_{sw}}$ is given by $\boldsymbol{S_{sw}}$, which is a subset of $\mathbb{R}^k$ of dimension $k-1$.

PS2d. Total Boundary. The total problem boundary is $\boldsymbol{S} = (\boldsymbol{S_{st}} \cup \boldsymbol{S_w} \cup \boldsymbol{S_{sw}}) \setminus ((\boldsymbol{S_{st}} \cap \boldsymbol{S_w}) \cup (\boldsymbol{S_{st}} \cap \boldsymbol{S_{sw}}) \cup (\boldsymbol{S_w} \cap \boldsymbol{S_{sw}}))$. That is, the set of boundary surfaces that belong to one and only one subdomain boundary.

PS3. Body Forces

PS3a. Gravitational Acceleration. The domain, $\boldsymbol{\Omega}$, is subjected to constant gravitational acceleration, which may be divided into components to account for the direction of the bottom of the domain relative to the radial direction from the centre of the Earth. When multiplied by the mass, this produces constant body forces on $\boldsymbol{\Omega}$.

PS3b. Seismic Excitation. The domain, $\boldsymbol{\Omega}$, may also be subjected to seismic excitation. This takes the form of a time varying acceleration vector. When multiplied by the mass, this produces time-varying body forces on $\boldsymbol{\Omega}$.

PS4. **Initial Conditions.** The initial conditions are the state of the system at the beginning of the simulation (typically when $t = 0$, but may be at a different time if the calculation is divided into multiple phases). For example, the initial pressure conditions might be defined by $p(\boldsymbol{\Omega}) = p_0$ where p_0 is a constant and the initial velocity conditions might be defined by $\dot{u}_i(\boldsymbol{\Omega}) = 0$. Solution in the time-domain requires specification of the initial state of all field variables.

PS5. **Boundary Conditions.** The boundary conditions, which may vary over time (*e.g.* due to interactions between subdomains), are divided into the following types.

- **PS5a.** **Kinematic (Dirichlet) Boundary Conditions.** This type of boundary condition involves specification of the primary solution variable along the boundary (*e.g.* $u_i(\boldsymbol{S}) = 0$).
- **PS5b.** **Natural (Neumann) Boundary Conditions.** This type of boundary condition involves specification of the stress conditions along the boundary (*e.g.* $\sigma_{ij}(\boldsymbol{S}) = 0$).
- **PS5c.** **Mixed Boundary Conditions.** This type of boundary condition means that portions of $\boldsymbol{S}$ are specified as in **PS5a** and other portions are specified as in **PS5b**, which would be the most typical case for DynSWS. Note that the different types of boundary condition should not overlap. For example, if one specifies displacement along a portion of the boundary, the stress must be determined by solving the differential equations and *vice versa*.

This concludes the description of the physical system that DynSWS is intended to model. Later in this document, Section 3.2 defines the set of mathematical models, data, and constraints involved in the solution.

3.1.4 Goal Statements

The major (functional) goals for DynSWS are as follows:

G1. Given the geometry, material properties, and time-varying loading for a system involving a structure, water, and soil, the software product should solve for the time-varying displacement, velocity, and acceleration fields over the problem domain.

G2. Given the displacement, velocity, and acceleration fields, the software product should determine the time-varying stress and strain fields over the problem domain.

3.2 Solution Characteristics Specification

3.2.1 Assumptions

This section summarizes the assumptions made in the theoretical and instanced models used to capture the class of problems that DynSWS is intended to solve. A brief description of the reasons for each assumption is also provided. This section is subdivided into three parts: general assumptions that apply either at the theoretical model or material behaviour level, assumptions for the 3-D instanced model, and assumptions for the 2-D instanced models.

Note that the 3-D model assumptions are presented first since the 2-D models generally begin with the 3-D model and make further simplifying assumptions.

3.2.1.1 General

This section details assumptions that apply generally either at the level of the theoretical models or material behaviour. This means that these assumptions apply equally for both the 2-D and 3-D models.

A1. The problem domain is isothermal. There is not a significant temperature gradient present. All material densities may be taken as constant (given the ambient temperature) and the energy balance due to temperature does not need to be considered. *Reason:* Temperature is not expected to contribute significantly to the solution. For special problems where the structure houses a process that generates heat, this assumption may be modified in the future.

A2. Relativistic effects can be ignored. Particle velocities are sufficiently less than the speed of light such that simple conservation of mass rather than the more general conservation of mass-energy may be employed. *Reason:* The actual particle velocities in the class of soil-water-structure interaction problems will not be greater than a very small fraction of the speed of light; it is not necessary to complicate the model by considering relativistic effects.

A3. There are no sources or sinks of material within any subdomain of the problem. Material may only enter or exit at the boundaries of the problem or transfer between subdomains. *Reason:* Practically speaking, it is not difficult to construct a model of any soil-water-structure interaction problem so that this assumption is satisfied. Therefore, there is no need to complicate the model by including the possibility of sources or sinks of water or other material inside the domain.

A4. Solids and fluids may be modelled as continua. Effects at the molecular scale or even at the scale of individual grains and pores (for porous materials such as soil) are ignored. An "infinitesimal volume" considers a region large enough that material properties may be averaged over a large number of molecules, grains, or pores while remaining small enough relative to the dimensions of the problem that it may be considered as a point. *Reason:* The scale of the soil-water-structure interaction problems under consideration is large enough compared to the size of an individual molecule, grain, or pore that such micro-scale interactions are not expected to influence the solution.

A5. There are no more than two material phases present in any subdomain. *Reason:* The main subdomains under consideration in a soil-water-structure interaction problem are water (*e.g.* reservoir), structural (*e.g.* concrete dam), and saturated soil (a mixture of soil particles and pore water). Consequently, a maximum of two simultaneous phases must be considered. This assumption may be modified at a later stage of development to accommodate unsaturated soils, which have some pore space filled with air.

A6. Materials are homogeneous in space and time within a specific subdomain. This means that the material properties do not vary over the subdomain. *Reason:* This assumption is relatively crude, particularly for soils, but typically detailed information about the spatial distribution of properties in the field is not available. Consequently, it is typical to assume properties that are averaged over the domain.

A7. Materials are isotropic within a specific subdomain. This means that the material properties do not depend on the orientation of the reference frame. *Reason:* This assumption is relatively accurate for the water and structure subdomains. However, for the soil subdomain it is not necessarily true. Despite this, as mentioned in **A6**, detailed information on the spatial distribution of material properties is typically not available so isotropy will be assumed for the soil subdomain as well. This assumption may be modified at a later development stage.

A8. The structural subdomain is impermeable. This means that there is no water phase in this region. The porosity of the structural subdomain is taken as zero and the influence of voids in the structural components is lumped in with the material properties of the solid. *Reason:* The permeability of the structural components is significantly lower than that of the underlying soil. Consequently, the effects of water permeating through the structure should be negligible.

A9. Strains in the solid components are less than ε_{SMALL}. Provided that strains are below this threshold, only first order terms must be considered in the definition for material deformation. *Reason:* This simplifies the constitutive law for the structural subdomain and the solid phase of the soil subdomain considerably. Note that this assumption precludes analysis at stress levels approaching failure. This assumption is likely to be modified at a later stage.

A10. The total stress in the structural subdomain is linearly related to the strain. This material model, commonly referred to as "linear elastic", means that no unrecoverable strains, such as plastic deformation or fracture, are induced by applied loads on the structure. *Reason:* The structure and its foundation are assumed to be designed such that the materials do not approach failure during regular operation or during small seismic events. Note that this assumption is primarily for the initial version of the model and is likely to be modified in the future.

A11. Water is an incompressible fluid. This means that there is no spatial density gradient in the water and the density of water is constant for a given temperature. *Reason:* In the context of soil-water-structure interaction problems, the compressibility of water is significantly lower than the compressibility of the solid materials (soil phase and structural components) to the extent that it may be ignored without influencing the solution. This is a common assumption in fluid dynamics [4, 5].

A12. Water is a Newtonian fluid. This means that the shear stress in water is linearly proportional to the strain rate (spatial velocity gradient) applied. *Reason:* This is the type of behaviour that most accurately reflects the behaviour of water and the behaviour that is typically assumed for water in fluid dynamics [4, 5].

A13. The flow of water is laminar, not turbulent. This means that water particles follow well-defined streamlines that do not cross. *Reason:* The flow velocities encountered in the class of problems that DynSWS is intended to solve are not expected to be large enough that turbulent flow will develop. The only region where such conditions may develop are in channels that allow water to pass through a dam to regulate reservoir levels and/or generate electricity, but DynSWS is not intended to model this portion of the system.

A14. Strain rates in the fluid components are less than $\dot{varepsilon}_{SMALL}$. Provided that strain rates are below this threshold, only first order terms must be considered in the definition for the rate of material deformation. *Reason:* This assumption is made for reasons similar to those in **A9**.

A15. The solid phase in a two phase subdomain is incompressible. This means that the actual soil particles do not change in density over time or space. *Reason:* It is expected that fluctuations in porosity at a point will dominate over fluctuations in material density.

A16. For porosity below n_{DENSE}, the behaviour of the soil phase of the soil-water subdomain is non-Newtonian. This means that when the soil phase is densely packed it does not follow the constitutive model for fluid implied by **A12** for water. The actual model for this regime may take a variety of forms, which will be addressed in a subsequent assumption. *Reason:* When soil becomes densely packed, this is the behaviour that has been observed in laboratory tests. See reference [6] and other references cited therein for more details.

A17. The effective stress in the soil phase is linearly related to the strain for porosity below n_{DENSE}. This is essentially the same as **A10** for the structure, but takes into account the fact that the soil behaviour is controlled by effective stress rather than total stress [7]. The total stress in the soil phase is the effective stress plus the isotropic pressure in the pore water. *Reason:* This is a relatively crude assumption for soil since its behaviour is non-linear except for very small strains. This is for the initial version of the model and is likely to be modified to an elasto-plastic or -viscoplastic model in the future.

A18. The soil phase behaves as a Newtonian fluid for porosity above n_{DENSE}. This means that for loosely packed soils, the constitutive model is essentially the same as that implied by **A12** for the water phase. *Reason:* When soil particles are loosely packed, the material behaviour depends on random contact between particles resulting in the same behaviour as a Newtonian fluid; see reference [6] for further details.

A19. A Cartesian coordinate system is adequate for describing the problem. This means that the coordinate system as taken as an orthogonal rectilinear reference frame. *Reason:* This is a temporary assumption to simplify the initial stage of development. Formulation and implementation in a cylindrical polar coordinate system is likely to be added at a later stage since it is more useful for problems with symmetry about a central axis such as the example shown in Figure 3.2.

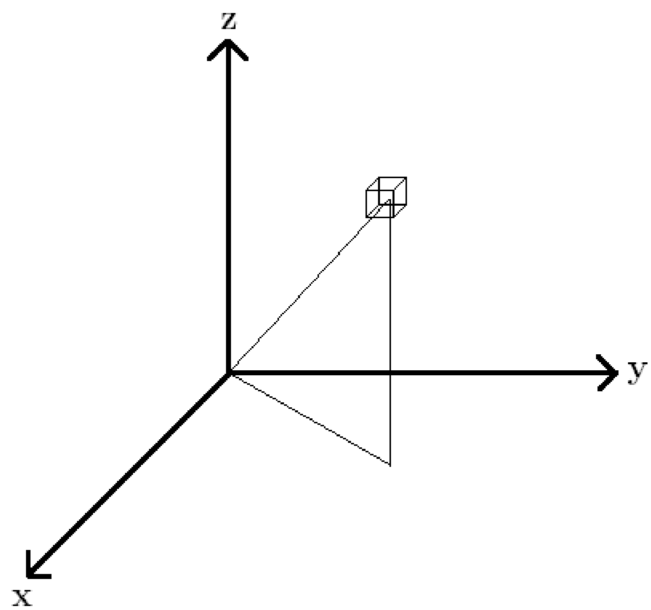

Figure 3.4: An infinitesimal volume of material located in a 3-D coordinate system

3.2.1.2 2-D Model

This section describes additional assumptions made in deriving the instanced models for the 2-D implementation.

A20. The system is subjected to conditions of plane strain. This means that strains in the out-of-plane direction are negligible and that the stress in the out-of-plane direction is a function of the in-plane stresses. This assumption requires that the longest dimension of the domain be at least $\alpha_p s$ times the shortest dimension. *Reason:* This assumption is appropriate for structures that are very long and the critical section is near the centre of the domain (far from the supports). Figure 3.1 shows an example of such a system. Modelling the system shown in 3.2 will require a different 2-D model.

3.2.2 Theoretical Model

This section describes the theoretical models that govern the class of problems that DynSWS is intended to solve. It serves well to present the underlying theory in its most general form to separate the statements of quantities that must be conserved from the instanced models for a particular material subdomain. Section 3.2.4 further refines the models and presents them in the form of the system of partial differential equations that DynSWS must solve for each region. The presentation of the theoretical models refer to the infinitesimal (with regard to **A4**) volume element shown in Figure 3.4 as defined by some appropriate coordinate system (*e.g.* Cartesian, cylindrical).

Making use of **A1** through **A5** the law of conservation of mass for a two phase system consisting of solid particles and a pore fluid is [4–6, 8, 9]:

TM1. Conservation of Mass for a Two Phase Substance

$$\frac{\partial\left(n\rho_f\right)}{\partial t}+\nabla\cdot\left(n\rho_f\dot{u}_i^f\right)=0 \tag{3.1}$$

$$\frac{\partial\left((1-n)\rho_s\right)}{\partial t}+\nabla\cdot\left((1-n)\rho_s\dot{u}_i^s\right)=0 \tag{3.2}$$

where n is the porosity of the solid particle phase, ρ is the material density of a given phase, t refers to time, $\dot{u}_i$ is the velocity vector of a given phase, ∇ is the del operator used in **TM1** with a dot product to represent the divergence (as defined for the coordinate system in use), and subscript/superscript f and s refer to the fluid and solid phases, respectively.

Also taking into account **A1** through **A5**, the law of conservation of momentum for a two phase system is [4–6, 8, 9]:

TM2. Conservation of Momentum for a Two Phase Substance

$$\frac{\partial\left(n\rho_f\dot{u}_i^f\right)}{\partial t}+\nabla\cdot\left(n\rho_f\dot{u}_i^f\dot{u}_i^f\right)=\nabla\cdot\sigma_{ij}^f+n\rho_f f_i+\beta\left(\dot{u}_i^s-\dot{u}_i^f\right) \tag{3.3}$$

$$\frac{\partial\left((1-n)\rho_s\dot{u}_i^s\right)}{\partial t}+\nabla\cdot\left((1-n)\rho_s\dot{u}_i^s\dot{u}_i^s\right)=\nabla\cdot\sigma_{ij}^s+(1-n)\rho_s f_i+\beta\left(\dot{u}_i^f-\dot{u}_i^s\right) \tag{3.4}$$

where σ_{ij} is the stress tensor, f_i represents applied accelerations (*e.g.* gravity, ground acceleration), β is a nonlinear porosity-dependent coefficient that accounts for momentum transfer between the two phases and all other terms are as defined previously. Note that **TM2** may be viewed as a statement of Newton's second law where the first term on the left-hand side accounts for the change in momentum at a point, the second term on the left-hand side accounts for the change in momentum due to the moving reference frame, and the other terms represent applied forces [4, 5].

Since water is assumed to behave as described in **A4**, **A11**, and **A12**, the stress at a point in a water phase is given by [5]:

TM3. Constitutive Model for a Newtonian Fluid

$$\sigma_{ij}=-p\delta_{ij}+\mu\dot{\varepsilon}_{ij} \tag{3.5}$$

where σ_{ij} is the total stress tensor, p is the thermodynamic pressure (isotropic, compression positive), δ_{ij} is the Kronecker delta, μ is the viscosity of water, $\dot{\varepsilon}_{ij}$ is the strain rate tensor, and subscripts ij are free indices in the style of tensor notation (*i.e.* i and j take on a range of values appropriate to the coordinate system in use). Taking into account **A14**, the strain rate tensor is defined as:

TM4. Strain Rate Tensor for Deformation of a Fluid

$$\dot{\varepsilon}_{ij}=\frac{\partial\dot{u}_i}{\partial x_j}+\frac{\partial\dot{u}_j}{\partial x_i} \tag{3.6}$$

where the terms $\dot{u}_i$ give the components of velocity and subscripts are as noted previously.

Since the structural components are assumed to behave as described in **A4**, **A6**, **A7**, **A9**, and **A10**, the stress at a point in the structure subdomain is given by [10]:

TM5. Constitutive Model for an Elastic Solid (Hooke's Law)

$$\sigma_{ij} = \frac{E}{3(1-2\nu)}\varepsilon_{kk}\delta_{ij} + \frac{E}{(1+\nu)}\left(\varepsilon_{ij} - \tfrac{1}{3}\varepsilon_{kk}\delta_{ij}\right) \tag{3.7}$$

where E is the elastic modulus, ν is Poisson's ratio, ε_{ij} is the strain tensor, repeated indices imply summation, and all other terms are as defined previously. Owing to **A9**, the definition of the strain tensor is:

TM6. Strain Tensor for Deformation of an Elastic Solid

$$\varepsilon_{ij} = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right) \tag{3.8}$$

where the terms u_i give the components of displacement and subscripts are as noted previously.

Considering **A16** and **A17**, the constitutive model for effective stress in the soil phase when $n < n_{DENSE}$ is:

TM7. Constitutive Model for an Elastic Soil (Hooke's Law, Effective Stress)

$$\sigma_{ij} = -p\delta_{ij} + \frac{E}{3(1-2\nu)}\varepsilon_{kk}\delta_{ij} + \frac{E}{(1+\nu)}\left(\varepsilon_{ij} - \tfrac{1}{3}\varepsilon_{kk}\delta_{ij}\right) \tag{3.9}$$

where σ_{ij} is the total stress in the solid phase, p is the pressure (compression positive) in the water phase, and all other terms are as defined previously. Considering **A18**, for $n \geq n_{DENSE}$, the constitutive model for the soil phase is the same as **TM3**, only differing in the value of μ.

This concludes the presentation of the theoretical models. Note that this section has presented the models in a very general form that may be specialized to handle a broad class of problems and subdomain types. Before refining the models for each subdomain it is important to define the data necessary to solve the problem, which is the intention of Section 3.2.3.

3.2.3 Data Definitions

This section describes all data necessary for DynSWS to understand and solve a problem involving dynamic soil-water-structure interaction. Note that the data definitions provided here are in abstract form, meaning that they do not imply the particular form the data will take when implemented in the computational model. Rather, the information in this section describes information that will need to be stored or computed in some manner to facilitate solution of the problem.

3.2.3.1 Coordinate System

This set of data provides the frame of reference for the geometry of the problem. Owing to **A19**, only a Cartesian coordinate system, as shown in Figure 3.5, will be considered at this stage. The Cartesian coordinate system is useful for problems such as that shown in Figure 3.1 where the geometry is rectilinear in nature. Coordinates are defined in terms of two or three linear dimensions [x,y,z] (depending on whether a 2-D or 3-D model is in use).

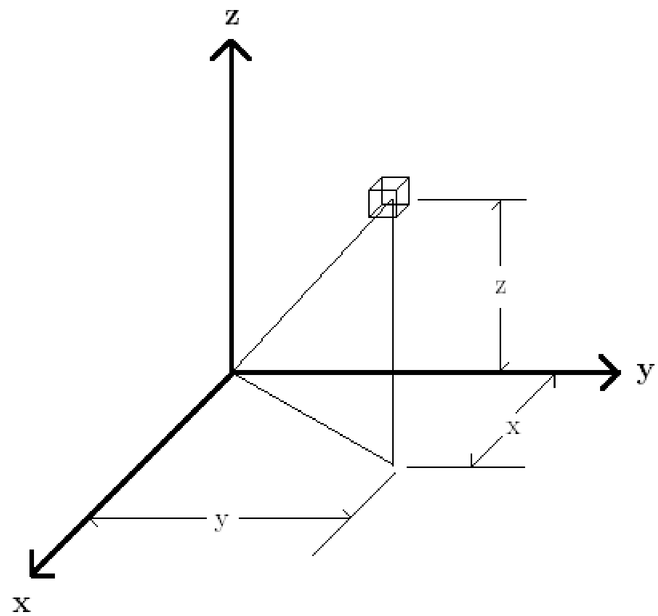

Figure 3.5: The Cartesian coordinate system

3.2.3.2 Problem Domain and Boundaries

Table 3.1 lists the data required to define the geometry of a soil-water-structure interaction problem. Each item will consist of a formatted set of real valued geometry data such as points and line segments out of which the problem domain is constructed. The format of this data will depend on the implementation of the solution algorithm

3.2.3.3 Material Properties

Table 3.2 gives the symbol, the name, and a brief description of the material properties required to define the behaviour described in Section 3.2.2. Note that all parameters in this section are real-valued scalars. Constraints on the data will be provided in Section 3.2.5.

3.2.3.4 Field Variables

Table 3.3 gives the symbol, the name, a brief description, and the type of the field variables required to describe and solve the theoretical model in Section 3.2.2. Note that all parameters in this section contain real-valued numbers. Constraints on the data will be provided in Section 3.2.5.

3.2.3.5 Sign Conventions

A number of terms in Table 3.3 require definition of a positive sign convention. This section aims to remove any ambiguity with regard to the sign convention used in DynSWS. All displacement, velocity, acceleration, and body force terms are taken as positive in the direction

Table 3.1: Definition of problem domain and boundary data

Symbol	Name	Units	Type
$\boldsymbol{\Omega}$	Total domain	L	Input
$\boldsymbol{\Omega_{st}}$	Structural subdomain	L	Input
$\boldsymbol{\Omega_{w}}$	Water subdomain	L	Input
$\boldsymbol{\Omega_{sw}}$	Soil subdomain	L	Input
$\boldsymbol{S}$	Total system boundary	L	Input
$\boldsymbol{S_{st}}$	Structural subdomain boundary	L	Input
$\boldsymbol{S_{w}}$	Water subdomain boundary	L	Input
$\boldsymbol{S_{sw}}$	Soil subdomain boundary	L	Input

Table 3.2: Definition of material property data

Symbol	Name	Units	Type
E	Elastic modulus	$M \cdot L^{-1} \cdot T^{-2}$	Input
n	Porosity	unitless	Input
β	Momentum transfer	$M \cdot L^{-3} \cdot T^{-1}$	Input
μ	Dynamic viscosity	$M \cdot L^{-1} \cdot T^{-1}$	Input
ν	Poisson's ratio	unitless	Input
ρ	Density	$M \cdot L^{-3}$	Input

Table 3.3: Definition of field variables

Symbol	Name	Units	Type
u_i	Displacement	L	Vector, Output
$\dot{u}_i$	Velocity	$L \cdot T^{-1}$	Vector, Output
$\ddot{u}_i$	Acceleration	$L \cdot T^{-2}$	Vector, Output
σ_{ij}	Stress	$M \cdot L^{-1} \cdot T^{-2}$	Tensor, Output
ε_{ij}	Strain	unitless	Tensor, Output
$\dot{\varepsilon}_{ij}$	Strain rate	T^{-1}	Tensor, Output
f_i	Body acceleration	$L \cdot T^{-2}$	Vector, Input
p	Pressure	$M \cdot L^{-1} \cdot T^{-2}$	Scalar, Output

of the coordinate axes as shown in Figure 3.5. The components of stress, σ_{ij}, strain, ε_{ij}, and strain rate, $\dot{\varepsilon}_{ij}$, are taken as positive in the directions shown in Figure 3.6. The isotropic pressure in the fluid phase (if present) is taken as positive in the direction indicated in Figure 3.7.

3.2.4 Instanced Model

This section refines the theoretical models introduced in Section 3.2.2 into forms appropriate for specific coordinate systems and specific subdomains (*i.e.* structure, water, and soil). Section 3.2.4.1 presents the form used for the 3-D model and Section 3.2.4.2 presents the form used for the 2-D model.

3.2.4.1 3-D Model

This section presents the instanced models used for the 3-D version of DynSWS. The main assumption made beyond what is assumed for the theoretical models is **A19**, regarding use of a Cartesian coordinate system. This implies that vector and tensor subscripts are in $\{x, y, z\}$. Sections 3.2.4.1.1, 3.2.4.1.2, and 3.2.4.1.3 present the instanced models for the structure, water, and soil subdomains, respectively.

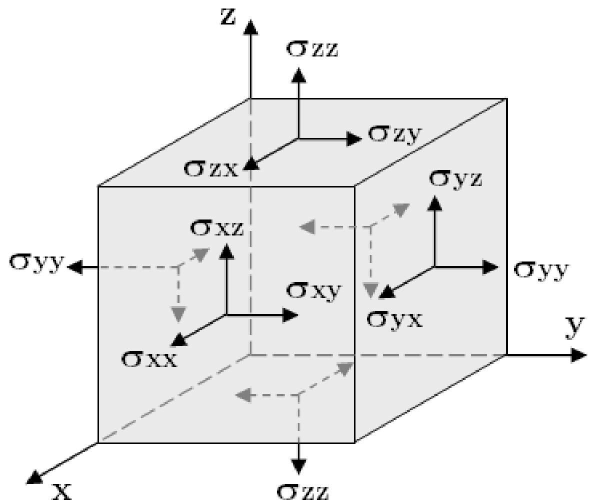

Figure 3.6: Positive sign convention for stress and strain (*Image Source:* [11])

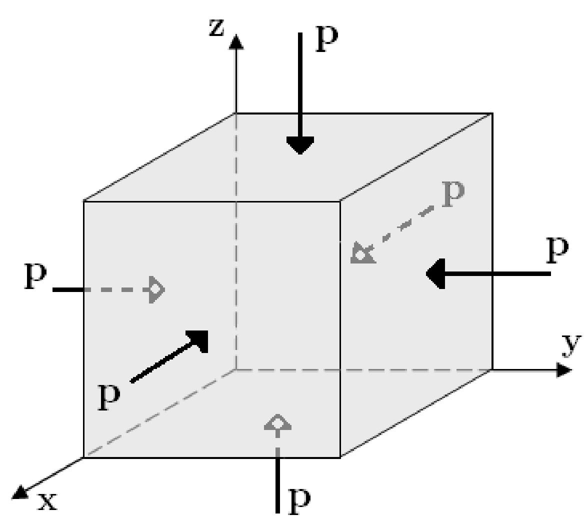

Figure 3.7: Positive sign convention for isotropic pressure

3.2.4.1.1 Structure Subdomain

Considering **TM1**, **A1**, **A6**, **A8**, **A9**, and **A10** mean that the density of the structural components is not changing over time or space and the material is not flowing. Consequently, **TM1** is satisfied identically for this subdomain and an instanced model is not required.

Considering **TM2**, **A1**, **A6**, and **A8** mean that density gradients can be ignored and that the porosity, n, can be taken as zero. Therefore, **TM2** reduces to a single set of three equilibrium equations given by:

IM1. Dynamic Equilibrium of Structure Subdomain

$$\rho_s \ddot{u}_i^s = \sigma_{ij,j}^s + \rho_s f_i \tag{3.10}$$

where repeated indices imply summation. Note that this represents 3 equations (one each for x,y,z directions) since i is a free index.

Taking into account **A19**, **A6**, and **A7** the stress tensor is **TM5** is symmetric. Consequently, it is convenient to express **TM5** as:

IM2. Matrix-Vector Form of Hooke's Law for an Elastic Solid

$$\begin{Bmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \sigma_{zz} \\ \sigma_{xy} \\ \sigma_{yz} \\ \sigma_{zx} \end{Bmatrix} = \frac{E}{(1+\nu)(1-2\nu)} \begin{bmatrix} 1-\nu & \nu & \nu & 0 & 0 & 0 \\ \nu & 1-\nu & \nu & 0 & 0 & 0 \\ \nu & \nu & 1-\nu & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{2}(1-2\nu) & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{2}(1-2\nu) & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{2}(1-2\nu) \end{bmatrix} \begin{Bmatrix} \varepsilon_{xx} \\ \varepsilon_{yy} \\ \varepsilon_{zz} \\ \gamma_{xy} \\ \gamma_{yz} \\ \gamma_{zx} \end{Bmatrix} \tag{3.11}$$

where it is understood that, although they are expressed as vectors, σ_{ij} and ε_{ij} must still obey the transformation rules of second order tensors. Similarly, **TM6** may be expressed as:

IM3. Matrix-Vector Form of Strain

$$\begin{Bmatrix} \varepsilon_{xx} \\ \varepsilon_{yy} \\ \varepsilon_{zz} \\ \gamma_{xy} \\ \gamma_{yz} \\ \gamma_{zx} \end{Bmatrix} = \begin{Bmatrix} \varepsilon_{xx} \\ \varepsilon_{yy} \\ \varepsilon_{zz} \\ 2\varepsilon_{xy} \\ 2\varepsilon_{yz} \\ 2\varepsilon_{zx} \end{Bmatrix} = \begin{bmatrix} \frac{\partial}{\partial x} & 0 & 0 \\ 0 & \frac{\partial}{\partial y} & 0 \\ 0 & 0 & \frac{\partial}{\partial z} \\ \frac{\partial}{\partial y} & \frac{\partial}{\partial x} & 0 \\ 0 & \frac{\partial}{\partial z} & \frac{\partial}{\partial y} \\ \frac{\partial}{\partial z} & 0 & \frac{\partial}{\partial x} \end{bmatrix} \begin{Bmatrix} u_x \\ u_y \\ u_z \end{Bmatrix} \tag{3.12}$$

Combining the 3 equations in **IM1**, the 6 equations in **IM2**, and the 6 equations in **IM3** gives 15 equations for 15 unknowns (3 displacement, 6 stress, and 6 strain) at a given point in time. The system is, in principle, amenable to solution in the temporal and spatial domains given appropriate initial and boundary conditions as defined in **PS4** and **PS5**, respectively.

3.2.4.1.2 Water Subdomain

Considering **TM1**, **A11** implies that the density of water does not change over time or space. Since this subdomain does not contain solids, n may be taken as one. Therefore, **TM1** reduces to a single scalar equation [4, 5]:

IM4. Conservation of Mass for an Incompressible Fluid

$$\dot{u}^f_{i,i} = 0 \tag{3.13}$$

Considering **TM2**, taking into account **A6**, **A7**, **A11**, **A12**, and **A13**, taking n as one and combining **TM2** with **TM3** and **TM4** gives the following set of 3 equations, commonly referred to as the Navier-Stokes equations [4, 5]:

IM5. Navier-Stokes Equations for Incompressible, Laminar Flow of a Newtonian Fluid

$$\rho_f(\ddot{u}^f_i + \dot{u}^f_j \dot{u}^f_{i,j}) = -p_{,i} + \rho_f f_i + \mu_f \dot{u}^f_{i,jj} \tag{3.14}$$

Taking **IM4** and **IM5** together gives 4 equations for 4 unknowns (3 velocity and 1 pressure) at a given point in time. Again, initial and boundary conditions must be specified as in **PS4** and **PS5**.

3.2.4.1.3 Soil Subdomain

Applying **A11** and **A15**, **TM1** reduces to a pair of scalar equations:

IM6. Conservation of Mass for Two Incompressible Phases

$$\dot{n} + n_{,i}\dot{u}^f_i + n\dot{u}^f_{i,i} = 0 \tag{3.15}$$

$$-\dot{n} - n_{,i}\dot{u}^s_i + (1-n)\dot{u}^s_{i,i} = 0 \tag{3.16}$$

where repeated indices imply summation.

By the same considerations taken for **IM1** and for **IM5**, but leaving the porosity, n, as a variable, the momentum balance for a two phase substance in Cartesian coordinates is:

IM7. Conservation of Momentum for a Two Phase Substance

$$n\rho_f(\ddot{u}^f_i + \dot{u}^f_j \dot{u}^f_{i,j}) = -p_{,i} + \rho_f f_i + \mu_f \dot{u}^f_{i,jj} + \beta(\dot{u}^s_i - \dot{u}^f_i) \tag{3.17}$$

$$(1-n)\rho_s(\ddot{u}^s_i + \dot{u}^s_j \dot{u}^s_{i,j}) = \sigma^s_{ij,j} + \rho_s f_i + \beta(\dot{u}^f_i - \dot{u}^s_i) \tag{3.18}$$

(3.19)

where σ^s_{ij} is given by **TM3** or **TM7** depending on the value of n.

From **IM6** and **IM7** there are a variable number of equations depending on the status of n at a point. For $n > n_{DENSE}$, both phases behave as Newtonian fluids yielding 8 equations and 8 unknowns (3 in $\dot{u}^f_i$, 3 in $\dot{u}^s_i$, p, and n). Otherwise, the pore water behaves as a Newtonian fluid and the solid particle phase behaves as an elastic solid (see **A17**) yielding 20 equations

(2 continuity, 6 momentum, 6 constitutive, and 6 strain-displacement) and 20 unknowns (3 in $\dot{u}_i^f$, 3 in u_i^s, 6 in σ_{ij}^s, 6 in ε_{ij}^s, p, and n). Again, it is possible, in principle, to solve this system given the appropriate initial conditions and boundary conditions as per **PS4** and **PS5**.

3.2.4.2 2-D Model

This section presents the instanced models for the 2-D version of DynSWS. The additional assumption made to obtain the 2-D version is **A20**, which corresponds to plane strain conditions. The former implies that vector and tensor subscripts are in $\{x, y\}$ and the latter implies that strains (and strain rates) in the z direction are zero. Sections 3.2.4.2.1, 3.2.4.2.2, and 3.2.4.2.3 present the instanced models for the structure, water, and soil subdomains, respectively.

3.2.4.2.1 Structure Subdomain

As for the 3-D model (described in Section 3.2.4.1.1), **TM1** is satisfied identically for the structure subdomain. Additionally, **IM1** is the same for the 2-D model with the restriction that $i, j \in \{x, y\}$. This means that the 2-D model cannot accommodate applied loads in the z direction, as one would expect.

Due to **A20**, **IM2** reduces to:

IM8. Matrix-Vector Form of Hooke's Law for an Elastic Solid for Plane Strain Conditions

$$\begin{Bmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \sigma_{xy} \end{Bmatrix} = \frac{E}{(1+\nu)(1-2\nu)} \begin{bmatrix} 1-\nu & \nu & 0 \\ \nu & 1-\nu & 0 \\ 0 & 0 & \frac{1}{2}(1-2\nu) \end{bmatrix} \begin{Bmatrix} \varepsilon_{xx} \\ \varepsilon_{yy} \\ \gamma_{xy} \end{Bmatrix} \tag{3.20}$$

Note that **A20** also implies the following about the stresses in the z direction:

IM9. Out-of-Plane Stresses for Plane Strain Conditions

$$\sigma_{zz} = \nu(\sigma_{xx} + \sigma_{yy}) \tag{3.21}$$

$$\sigma_{yz} = \sigma_{zx} = 0 \tag{3.22}$$

Similarly, **A20** reduces **IM3** to:

IM10. Matrix-Vector Form of Strain for Plane Strain Conditions

$$\begin{Bmatrix} \varepsilon_{xx} \\ \varepsilon_{yy} \\ \gamma_{xy} \end{Bmatrix} = \begin{Bmatrix} \varepsilon_{xx} \\ \varepsilon_{yy} \\ 2\varepsilon_{xy} \end{Bmatrix} = \begin{bmatrix} \frac{\partial}{\partial x} & 0 \\ 0 & \frac{\partial}{\partial y} \\ \frac{\partial}{\partial y} & \frac{\partial}{\partial x} \end{bmatrix} \begin{Bmatrix} u_x \\ u_y \end{Bmatrix} \tag{3.23}$$

Combining the 2 equations in **IM1** for 2-D, the 3 equations in **IM8**, and the 3 equations in **IM10** gives 8 equations for 8 independent unknowns (2 displacement, 3 stress, and 3 strain). Note that **IM9** gives an additional equation for σ_{zz}, which is dependent on σ_{xx} and σ_{yy}.

3.2.4.2.2 Water Subdomain

The water subdomain in the 2-D model uses **IM4** and **IM5** from the 3-D model with **A20** implying that $i, j \in \{x, y\}$. For the 2-D case these models yield 3 equations for 3 unknowns (2 velocity and 1 pressure).

3.2.4.2.3 Soil Subdomain

Again, **A20** allows the 3-D models for the soil subdomain (**IM6** and **IM7**) to be reused with the subscripts $i, j \in \{x, y\}$. For the case that $n > n_{DENSE}$, **IM6** and **IM7** yield 6 equations for 6 unknowns (2 in $\dot{u}_i^f$, 2 in $\dot{u}_i^s$, p, and n) at any point in the subdomain. Otherwise, the solid phase behaves according to **A17** and **IM6**, **IM7**, **TM6**, and **TM7** yield 12 equations (2 continuity, 4 momentum, 3 constitutive, and 3 strain-displacement) for 12 unknowns (2 in $\dot{u}_i^f$, 2 in u_i^s, 3 in σ_{ij}^s, 3 in ε_{ij}^s, p, and n).

3.2.5 Data Constraints

This section provides information on the physical constraints and software system constraints on the solution data. Section 3.2.5.1 gives constraints on the problem domain data, Section 3.2.5.2 gives constraints on material properties, and Section 3.2.5.3 gives constraints on field variables.

3.2.5.1 Problem Domain and Boundaries

Table 3.4 gives the symbol, units, physical constraints, and system constraints for each of the material property variables introduced in Table 3.1.

3.2.5.2 Material Properties

Table 3.5 gives the symbol, units, physical constraints, and system constraints for each of the material property variables introduced in Table 3.2.

3.2.5.3 Field Variables

Table 3.6 gives the symbol, units, physical constraints, and system constraints for each of the field variables introduced in Table 3.3.

Table 3.4: Constraints on problem domain and boundary data

Symbol	Physical Constraints	System Constraints
$\boldsymbol{\Omega}$	$-\infty < \boldsymbol{\Omega} < \infty$	$\boldsymbol{\Omega_{MIN}} \leq \boldsymbol{\Omega} \leq \boldsymbol{\Omega_{MAX}}$
$\boldsymbol{\Omega_{st}}$	$-\infty < \boldsymbol{\Omega_{st}} < \infty$	$\boldsymbol{\Omega_{MIN}} \leq \boldsymbol{\Omega_{st}} \leq \boldsymbol{\Omega_{MAX}}$
$\boldsymbol{\Omega_{w}}$	$-\infty < \boldsymbol{\Omega_{w}} < \infty$	$\boldsymbol{\Omega_{MIN}} \leq \boldsymbol{\Omega_{w}} \leq \boldsymbol{\Omega_{MAX}}$
$\boldsymbol{\Omega_{sw}}$	$-\infty < \boldsymbol{\Omega_{sw}} < \infty$	$\boldsymbol{\Omega_{MIN}} \leq \boldsymbol{\Omega_{sw}} \leq \boldsymbol{\Omega_{MAX}}$
$\boldsymbol{S}$	$-\infty < \boldsymbol{S} < \infty$	$\boldsymbol{S_{MIN}} \leq \boldsymbol{S} \leq \boldsymbol{S_{MAX}}$
$\boldsymbol{S_{st}}$	$-\infty < \boldsymbol{S_{st}} < \infty$	$\boldsymbol{S_{MIN}} \leq \boldsymbol{S_{st}} \leq \boldsymbol{S_{MAX}}$
$\boldsymbol{S_{w}}$	$-\infty < \boldsymbol{S_{w}} < \infty$	$\boldsymbol{S_{MIN}} \leq \boldsymbol{S_{w}} \leq \boldsymbol{S_{MAX}}$
$\boldsymbol{S_{sw}}$	$-\infty < \boldsymbol{S_{sw}} < \infty$	$\boldsymbol{S_{MIN}} \leq \boldsymbol{S_{sw}} \leq \boldsymbol{S_{MAX}}$

Table 3.5: Constraints on material property data

Symbol	Physical Constraints	System Constraints
E	$0 \leq E < \infty$	$E_{MIN} \leq E \leq E_{MAX}$
n	$0 \leq n \leq 1$	$n_{MIN} \leq n \leq n_{MAX}$
β	$0 \leq \beta < \infty$	$\beta_{MIN} \leq \beta \leq \beta_{MAX}$
μ	$0 \leq \mu < \infty$	$\mu_{MIN} \leq \mu \leq \mu_{MAX}$
ν	$0 \leq \nu \leq 0.5$	$\nu_{MIN} \leq \nu \leq \nu_{MAX}$
ρ	$0 \leq \rho_f < \infty$	$\rho_{MIN} \leq \rho \leq \rho_{MAX}$

Table 3.6: Constraints on field variables

Symbol	Physical Constraints	System Constraints
u_i	$-\infty < u_i < \infty$	$u_{MIN} \leq u_i \leq u_{MAX}$
$\dot{u}_i$	$-\infty < \dot{u}_i < \infty$	$\dot{u}_{MIN} \leq \dot{u}_i \leq \dot{u}_{MAX}$
$\ddot{u}_i$	$-\infty < \ddot{u}_i < \infty$	$\ddot{u}_{MIN} \leq \ddot{u}_i \leq \ddot{u}_{MAX}$
σ_{ij}	$-\infty < \sigma_{ij} < \infty$	$\sigma_{MIN} \leq \sigma_{ij} \leq \sigma_{MAX}$
ε_{ij}	$-\infty < \varepsilon_{ij} < \infty$	$\varepsilon_{MIN} \leq \varepsilon_{ij} \leq \varepsilon_{MAX}$
$\dot{\varepsilon}_{ij}$	$-\infty < \dot{\varepsilon}_{ij} < \infty$	$\dot{\varepsilon}_{MIN} \leq \dot{\varepsilon}_{ij} \leq \dot{\varepsilon}_{MAX}$
f_i	$-\infty < f_i < \infty$	$f_{MIN} \leq f_i \leq f_{MAX}$
p	$0 < p < \infty$	$p_{MIN} \leq p \leq p_{MAX}$

3.2.6 System Behaviour

This section details the behaviour of the DynSWS system under various operating conditions. The general workflow will be: read input file(s), perform calculations, generate output file(s). The following are the details of these operations:

SB1. Read Input. On activation, DynSWS should attempt to read the input file(s) with the given file name.

- **SB1a. Missing input file(s).** If one or more required input files are missing, DynSWS should place an appropriate error message into the output data structure and proceed with output generation (go to **SB3**).
- **SB1b. Missing input data or wrong file format.** If the input files cannot be read because data is missing or the file format is incorrect, DynSWS should place an appropriate error message into the output data structure and proceed with output generation (go to **SB3**).
- **SB1c. Input data violates constraint(s).** If the data is successfully read from the input file(s), but some of the data violates the physical or system constraints, DynSWS should place an appropriate error message into the output data structure and proceed with output generation (go to **SB3**).
- **SB1d. All input file(s) present, all data present, and all data satisfies constraints.** If the input data set satisfies all requirements, DynSWS should place the input data into the calculation data structures and proceed with calculation (go to **SB2**).

SB2. Perform Calculation. Following the reading of input data, DynSWS should proceed with calculation of the solution over the specified spatial and temporal domains.

- **SB2a. Calculation fails because solution variable(s) violate constraint(s).** If the calculation procedure stops because one or more of the solution variables have reached a physical or system constraint, DynSWS should stop the calculation, place a relevant subset of the output and an appropriate error message into the output data structures, and proceed with output generation (go to **SB3**).
- **SB2b. Calculation fails because algorithm fails to converge.** If the solution procedure fails to converge to a solution, DynSWS should stop calculation, place a relevant subset of the output and an appropriate error message into the output data structures, and proceed with output generation (go to **SB3**). Note that convergence will be documented at a later stage along with the numerical algorithm. The numerical algorithm will be specified at the design stage (see reference [12]) and documented at the interface specification stage (see reference [13]).
- **SB2c. Calculation completes successfully.** If the calculation phase completes without violating any data constraints or failing to converge, DynSWS should place the results into the output data structures and proceed with output generation (go to **SB3**).

SB3. Generate Output. After completion of calculation (or a stop due to an error), DynSWS should generate the necessary output file(s).

- **SB3a. Return error message(s).** If any error messages have been generated, DynSWS should place this information into the output file(s).
- **SB3b. No error messages generated.** If **SB1** and **SB2** complete without generating any error messages, DynSWS should place this information into the output file(s).
- **SB3c. Return calculation results.** If **SB2** returned partial or full results, DynSWS should place this information into the output file(s).

SB4. Stop. After completion of **SB3**, DynSWS is finished and should stop.

3.3 Non-Functional Requirements

This section defines the system requirements that are in addition to the technical requirements of solving soil-water-structure interaction problems.

3.3.1 Accuracy of Input Data

The class of problems that DynSWS is intended to solve exist in the real-world and involve the behaviour of large-scale systems of man-made and natural materials. It should be noted when applying the model that much of the input data is subject to sources of error that are impossible for DynSWS to predict (not even in principle).

The geometry of a structure such as a dam is very difficult to obtain precisely. If dimensions are taken from design drawings, one must be aware that the constructed dimensions may be different (and perhaps more relevant). On the other hand, if dimensions are taken from survey data of the structure in the field, there may exist systematic error in the measuring technique and/or error due to mistakes made by human operators.

Material properties can also be very difficult to obtain accurately. Often values that are obtained under controlled laboratory conditions do not match very well with *in situ* values (for both natural materials, such as soil, and man-made materials, such as concrete). Even if the material properties obtained in the laboratory accurately represent the values for the sampling location, the properties may have significant spatial variability.

Actual ground accelerations during a seismic event are difficult to predict. Recorded data are often not from the actual location of the structure being considered. Furthermore, measured historical data does not necessarily correspond to events that may take place in the future. Since differences in the amplitude and frequency content of the ground vibrations may have considerable influence on the results (see reference [14] for more details), one should not rely on a single set of input loads when analysing the response of a soil-water-structure system to seismic behaviour.

Since DynSWS will be unable to verify whether geometry, material, or loading inputs correspond well with the real-world values, it must be left to the user to ensure the accuracy of the values provided to the program. DynSWS is not intended to be used as a tool to solve a particular problem in a single run with a single set of input data. It is important that the user carry out sensitivity analysis by varying the input values for parameters that they are uncertain of to arrive at an appreciation for the general behaviour of the system rather than relying on a single set of results.

3.3.2 Sensitivity of the Model

It is not known *a priori* how sensitive DynSWS will be in the general case. Consequently, the responsibility of checking the sensitivity of the results to changes in input will typically be left to the user on a case-by-case basis. That being said, DynSWS should be packaged with some documentation of the sensitivity under some standard test cases.

NFR1. DynSWS should be bundled with documentation indicating the sensitivity of the model for the test cases documented in **NFR4**.

3.3.3 Tolerance of the Solution

The solution algorithm used in DynSWS is likely to involve a time stepping procedure containing an iterative solver that runs on each time step. The following are the requirements for these components:

NFR2. On a given time step, the iterative solver in DynSWS should proceed until the approximation error (the definition of this is to be defined at the implementation stage in light of the actual numerical algorithm) falls below $\epsilon_{CONVERGE}$.

NFR3. On a given time step, the iterative solver should achieve **NFR2** within N_{MAX} iterations.

3.3.4 Solution Validation Strategies

Although it will not be possible to validate DynSWS exhaustively due to the infinite variability in problems that might be supplied as input, it is still possible to compare the results for some standard test cases that have either an analytical or well accepted numerical solution.

NFR4. DynSWS should be validated against the following simplified test cases:

- **NFR4a.** Static solution for a structure supported by soil and subjected to hydrostatic pressure.
- **NFR4b.** One-dimensional laminar flow in a channel.
- **NFR4c.** Dynamic solution for a cantilever-type dam structure subjected to seismic excitation at the base.

The author's thesis research will likely involve laboratory testing of physical models of soil-water-structure interaction problems and investigation of the response of real structures to seismic excitation. These should also be taken into account in validating the model implemented in DynSWS.

NFR5. DynSWS should be validated against the results of laboratory tests performed on physical models of soil-water-structure interaction problems.

NFR6. DynSWS should be validated against field measurements of the response of soil-water-structure systems to seismic excitation.

3.3.5 Look-and-Feel Requirements

Given that users will not directly interact with DynSWS (see Figure 2.1), it should appear to operate as a "black box" from the perspective of the user (or driver program). That is, activation of the model should be through a single file/function/subroutine. This will reduce the complexity of interactions with other software products.

NFR7. The software product should exist as a single compiled executable file or shared library that exports a single function or subroutine to be called by a driver program.

3.3.6 Usability Requirements

Use of DynSWS by the pre-processor or driver program should not require knowledge of the implementation beyond the required input data and the format of the input/output files.

NFR8. A driver program should be able to activate DynSWS by calling a function of the form *DynSWS (infile, outfile)* where *infile* is the name of the file containing input to the model and *outfile* is the name of the file containing the model outputs (including error messages). The only knowledge of the implementation required should be the format of *infile* for the pre-processor and *outfile* for the post-processor.

3.3.7 Performance Requirements

Given that DynSWS is intended to solve a more comprehensive problem set than most available software packages of a similar nature, improved performance is not a strict requirement. However, DynSWS should aim to achieve comparable computation times to existing products for a standard set of test cases.

NFR9. DynSWS should be able to compute the solution to the test cases in **NFR4** in a time within $r_{comptime}\%$ of the time required by Plaxis2D Dynamics [15], Plaxis3D Dynamics [16], and/or OpenFOAM [17] as appropriate.

3.3.8 Maintainability Requirements

The first of the non-functional requirements for maintainability derives from the fact that the 2-D/3-D models in Cartesian/cylindrical coordinates form a family of programs that are likely to have many similar members. It is not desirable for a change in a shared portion of the algorithm to require updates to multiple instances of essentially the same procedure.

NFR10. Wherever a portion of the code (function or subroutine) is used by both 2-D and 3-D implementations and/or different coordinate systems, there should be a single version implemented in a manner abstract enough that all versions are obtained by varying only the input parameters.

The second and third non-functional requirements for maintainability derive from the fact that the format of the input and output files is highly likely to change as the model develops. The only portions of the DynSWS algorithm that should require knowledge of these formats are the functions/subroutines that read from or write to the input/output files.

NFR11. The code should be designed in such a way that a change in the format of the input file can be accommodated by modifying only one function or subroutine; the same should be true for a change in the format of the output file.

NFR12. Internal data structures required in the solution of the problem should not depend on the format of the input/output files. Changes to the format of these files should not require changes to the data structures in the solution algorithm.

3.3.9 Portability Requirements

The non-functional requirement for portability derives from the fact that the commercial software maintained by the potential industry partner runs on multiple operating systems. This partner would require that DynSWS be capable of running on any operating system supported by their commercial code.

NFR13. The software product should run on both Windows and Macintosh operating systems.

3.3.10 Security Requirements

DynSWS will not handle any sensitive information about its users. Consequently, from the user perspective there is no foreseeable security concern. However, from the perspective of the author, future developers, and the potential industry partner, the source code of the implementation represent intellectual property that should not be accessible to end users.

NFR14. Versions of DynSWS that are distributed for use by persons other than the developer(s) should not contain or expose portions of the source code deemed as the intellectual property of the developer(s).

4 Other System Issues

This section discusses additional items that are not necessarily involved in the current design plans for DynSWS, but may have some influence on the implementation and future development of the software product. Section 4.1 exposes the author's thoughts on uncertain issues that may influence the specifications for DynSWS at a later stage. Section 4.2 provides a list of existing software packages that may be capable of solving the type of problem that DynSWS is designed for. Section 4.3 lists problems that have arisen at the implementation stage that will affect the specification of DynSWS, but have not yet been addressed. Section 4.4 serves as a laundry list of ideas for features of future versions of DynSWS that are not included in the current specification.

4.1 Open Issues

The modelling of simultaneous interactions of soil-water-structure systems is an area of current research in geotechnical and structural engineering. As such, there is uncertainty in the ability of existing modelling frameworks to capture these interactions. At this stage, a number of simplifying assumptions have been made in the specification of DynSWS that may affect the software products' ability to model certain problems (*e.g.* problems where deformations approach the level of failure, problems that are best modelled using symmetry about a central axis). This aspect is likely to result in revisions, at least to the details, of the theoretical/instanced models specified for DynSWS.

The theoretical models proposed for DynSWS, while solvable in principle, are intractable for all but the simplest problem definitions. Since DynSWS is required to solve real world problems, the implementation will undoubtedly use numerical approximations to the theoretical/instanced models proposed herein. The introduction of numerical approximation obviously introduces a level of uncertainty into the results. This may be unavoidable, but it is not clear at this stage how much it will influence the production of accurate results.

The various subdomains of the class of soil-water-structure interaction problems that DynSWS is intended to solve are governed by a related, but slightly different, set of differential equations. There is uncertainty related to the procedure for making the mathematical models cooperate. Facilitating this cooperation may require revision of the proposed theoretical models for DynSWS at a later stage. There is related uncertainty involved in making the numerical approximation techniques for the different subdomains cooperate.

4.2 Off-the-Shelf Solutions

A number of software packages exist that are designed to solve systems defined by the type of partial differential equations involved in the specification of DynSWS. While this list is not exhaustive, here are some packages that exist on the market that are likely capable of solving at least a subset of the soil-water-structure interaction class of problems:

- ADINA - Automatic Dynamic Incremental Nonlinear Analysis [18] (commercial FEA package)

- ANSYS - Structural Dynamics, Multiphysics, Fluid Dynamics, Explicit Dynamics [19] (commercial FEA/CFD package)
- OpenFOAM - The open source CFD (Computational Fluid Dynamics) toolbox [17] (open source CFD package)
- Plaxis - Essential software for geotechnical professionals [20] (commercial geotechnical FEA package)

4.3 New Problems

DynSWS is not yet at the implementation stage. Problems arising in the implementation that affect the specification of DynSWS will be listed here.

4.4 Waiting Room

This section lists features that are likely to be added to the DynSWS specification at a later stage. The documentation in this section is necessarily brief since it is just a list of ideas that will be expanded on later.

- Determine the constraints placed on the format of input and output files for compatibility with the pre- and post-processor.
- Formulate instanced models for cylindrical coordinates in 2-D and 3-D. This will be useful for modelling problems with a central axis of symmetry.
- Include elasto-plastic or -viscoplastic models into material behaviour on Ω_{st} and Ω_{sw}. This will be necessary for modelling the response of soil-water-structure systems as failure strains are approached.
- Include non-mass-proportional dynamic loading in theoretical/instanced models. While dynamic loading due to seismic excitation can be captured by mass proportional forces, modelling the response to other types of dynamic loading (*e.g.* blast loading) will require arbitrary time-varying applied loads.

5 Traceability Matrix

Figure 5.1: Traceability Matrix, Part 1 of 3

Goal	Physical System	Data / Model	Assumption									
			A1	**A2**	**A3**	**A4**	**A5**	**A6**	**A7**	**A8**	**A9**	**A10**
G1	PS1b, PS1c, PS1d	**TM1**	X	X	X	X	X	X				
		TM2	X	X	X	X	X	X	X			
G2	PS1c	**TM3**		X		X		X	X		X	
		TM4		X		X					X	
	PS1b	**TM5**				X		X	X	X	X	X
	PS1b, PS1d	**TM6**				X					X	
	PS1d	**TM7**				X	X	X	X		X	
G1, G2	PS1b	**IM1**	X		X	X		X	X	X		X
G2		**IM2**				X		X	X	X	X	X
		IM3				X					X	
G1	PS1c	**IM4**	X	X	X	X		X				
G1, G2		**IM5**	X	X	X	X		X	X			
G1	PS1d	**IM6**	X	X	X	X	X	X	X			
G1, G2		**IM7**	X	X	X	X	X	X	X			
G2	PS1b	**IM8**				X		X	X		X	X
		IM9				X		X			X	X
		IM10				X					X	
G1, G2	PS1a	**2D**										
		3D										
		M,L,T										
G2	PS1b, PS1d	**E**				X		X	X			X
	PS1d	$\mathbf{n}$			X	X	X					
		$\boldsymbol{\beta}$		X		X	X					
	PS1c, PS1d	$\boldsymbol{\mu_f}$				X		X	X			
	PS1d	$\boldsymbol{\mu_s}$				X		X	X			
	PS1b, PS1d	$\boldsymbol{\nu}$				X		X	X			X
G1, G2	PS1c, PS1c	$\boldsymbol{\rho_f}$	X		X	X		X	X			
	PS1b, PS1d	$\boldsymbol{\rho_s}$	X		X	X		X	X			
G1	PS1c, PS1d	$\mathbf{u_i^f}$, $\mathbf{\dot{u}_i^f}$, $\mathbf{\ddot{u}_i^f}$				X						
	PS1b, PS1d	$\mathbf{u_i^s}$, $\mathbf{\dot{u}_i^s}$, $\mathbf{\ddot{u}_i^s}$				X						
G2		$\boldsymbol{\sigma_{ij}}$				X					X	X
	PS1b, PS1d	$\boldsymbol{\varepsilon_{ij}}$				X					X	X
	PS1c, PS1d	$\boldsymbol{\dot{\varepsilon}_{ij}}$		X		X					X	
	PS1b, PS1c, PS1d	$\mathbf{f_i}$										
	PS1c, PS1d	$\mathbf{p}$				X	X					

Figure 5.2: Traceability Matrix, Part 2 of 3

Goal	Physical System	Data / Model	Assumption									
			A11	**A12**	**A13**	**A14**	**A15**	**A16**	**A17**	**A18**	**A19**	**A20**
G1	PS1b, PS1c, PS1d	**TM1**										
		TM2			X							
G2	PS1c	**TM3**		X					X			
		TM4										
	PS1b	**TM5**										
	PS1b, PS1d	**TM6**										
	PS1d	**TM7**					X	X				
G1, G2	PS1b	**IM1**										
G2		**IM2**										
		IM3										
G1	PS1c	**IM4**	X									
G1, G2		**IM5**	X	X	X							
G1	PS1d	**IM6**	X			X						
G1, G2		**IM7**	X	X	X	X	X	X	X			
G2	PS1b	**IM8**					X	X				X
		IM9										X
		IM10										X
G1, G2	PS1a	**2D**									X	
		3D								X		
		M,L,T										
G2	PS1b, PS1d	$\mathbf{E}$						X				
	PS1d	$\mathbf{n}$	X			X	X					
		$\boldsymbol{\beta}$					X					
	PS1c, PS1d	$\boldsymbol{\mu_f}$		X								
	PS1d	$\boldsymbol{\mu_s}$					X		X			
	PS1b, PS1d	$\boldsymbol{\nu}$						X				
G1, G2	PS1c, PS1c	$\boldsymbol{\rho_f}$	X									
	PS1b, PS1d	$\boldsymbol{\rho_s}$				X						
G1	PS1c, PS1d	$\mathbf{u_i^f}$, $\mathbf{\dot{u}_i^f}$, $\mathbf{\ddot{u}_i^f}$								X	X	
	PS1b, PS1d	$\mathbf{u_i^s}$, $\mathbf{\dot{u}_i^s}$, $\mathbf{\ddot{u}_i^s}$					X			X	X	
G2		$\boldsymbol{\sigma_{ij}}$					X	X	X	X	X	X
	PS1b, PS1d	$\boldsymbol{\varepsilon_{ij}}$						X		X	X	X
	PS1c, PS1d	$\boldsymbol{\dot{\varepsilon}_{ij}}$							X	X	X	
	PS1b, PS1c, PS1d	$\mathbf{f_i}$								X	X	
	PS1c, PS1d	$\mathbf{p}$								X	X	

Figure 5.3: Traceability Matrix, Part 3 of 3

Goal	Physical System	Data / Model	Instanced Model									
			IM1	**IM2**	**IM3**	**IM4**	**IM5**	**IM6**	**IM7**	**IM8**	**IM9**	**IM10**
G1	PS1b, PS1c, PS1d	**TM1**				X		X				
		TM2	X				X		X			
G2	PS1c	**TM3**					X		X			
		TM4					X		X			
	PS1b	**TM5**	X	X						X	X	
	PS1b, PS1d	**TM6**	X		X				X			X
	PS1d	**TM7**							X			
G1, G2	PS1b	**IM1**	X									
G2		**IM2**		X								
		IM3			X							
G1	PS1c	**IM4**				X	X					
G1, G2		**IM5**					X					
G1	PS1d	**IM6**						X	X			
G1, G2		**IM7**							X			
G2	PS1b	**IM8**							X	X		
		IM9									X	
		IM10							X			X
G1, G2	PS1a	**2D**								X	X	X
		3D	X	X	X	X	X	X	X			
		M,L,T	X	X	X	X	X	X	X	X	X	X
G2	PS1b, PS1d	$\mathbf{E}$		X					X	X		
	PS1d	$\mathbf{n}$						X	X			
		$\boldsymbol{\beta}$							X			
	PS1c, PS1d	$\boldsymbol{\mu_f}$					X		X			
	PS1d	$\boldsymbol{\mu_s}$							X			
	PS1b, PS1d	$\boldsymbol{\nu}$		X					X	X	X	
G1, G2	PS1c, PS1c	$\boldsymbol{\rho_f}$					X	X	X			
	PS1b, PS1d	$\boldsymbol{\rho_s}$						X	X			
G1	PS1c, PS1d	$\mathbf{u_i^f}, \mathbf{\dot{u}_i^f}, \mathbf{\ddot{u}_i^f}$				X	X	X	X			
	PS1b, PS1d	$\mathbf{u_i^s}, \mathbf{\dot{u}_i^s}, \mathbf{\ddot{u}_i^s}$	X		X			X	X			X
G2		$\boldsymbol{\sigma_{ij}}$	X	X			X		X	X	X	
	PS1b, PS1d	$\boldsymbol{\varepsilon_{ij}}$	X	X	X				X	X		X
	PS1c, PS1d	$\boldsymbol{\dot{\varepsilon}_{ij}}$					X		X			
	PS1b, PS1c, PS1d	$\mathbf{f_i}$	X				X		X			
	PS1c, PS1d	$\mathbf{p}$					X		X			

Brandon Karchewski
McMaster University - Dept. of Civil Engineering

6 List of Possible Changes in the Requirements

General ideas about changes to the specification for DynSWS are documented in other sections of the report. In particular, see notes to the model assumptions (Section 3.2.1), open issues (Section 4.1), new problems (Section 4.3), and the waiting room (Section 4.4). More detailed statements of the affected assumptions and models as well as mathematical statements of proposed changes will be placed in this section at a later time.

7 Values of Auxiliary Constants

Table 7.1: Values of auxiliary constants for domain and boundary data

Symbol	Value	Note
$\boldsymbol{\Omega_{MIN}}$	-10^{11}	
$\boldsymbol{\Omega_{MAX}}$	10^{11}	
$\boldsymbol{S_{MIN}}$	-10^{11}	
$\boldsymbol{S_{MAX}}$	10^{11}	

Table 7.2: Values of auxiliary constants for numerical implementation

Symbol	Value	Note
$\epsilon_{CONVERGE}$	10^{-4}	
N_{MAX}	500	

Table 7.3: Values of auxiliary constants for material properties

Symbol	Value	Note
E_{MIN}	0	
E_{MAX}	10^{11}	
n_{DENSE}	0.5	See reference [6]
n_{MIN}	10^{-5}	Use single phase solid model below this value
n_{MAX}	0.99999	Use single phase fluid model above this value
β_{MIN}	0	
β_{MAX}	10^{11}	This value is not certain at this point
μ_{MIN}	0	
μ_{MAX}	10^{11}	
ν_{MIN}	0	
ν_{MAX}	0.499	Possible numerical issues above this value
ρ_{MIN}	0	
ρ_{MAX}	10^{11}	

Table 7.4: Values of auxiliary constants for field variables

Symbol	Value	Note
f_{MIN}	-10^{11}	
f_{MAX}	10^{11}	
p_{MIN}	-10^{11}	
p_{MAX}	10^{11}	
u_{MIN}	-10^{11}	
u_{MAX}	10^{11}	
$\dot{u}_{MIN}$	-10^{11}	
$\dot{u}_{MAX}$	10^{11}	
$\ddot{u}_{MIN}$	-10^{11}	
$\ddot{u}_{MAX}$	10^{11}	
σ_{MIN}	-10^{11}	
σ_{MAX}	10^{11}	
ε_{SMALL}	10^{-2}	
ε_{MIN}	$-\varepsilon_{SMALL}$	
ε_{MAX}	ε_{SMALL}	
$\dot{\varepsilon}_{SMALL}$	10^{-2}	
$\dot{\varepsilon}_{MIN}$	$-\dot{\varepsilon}_{SMALL}$	
$\dot{\varepsilon}_{MAX}$	$\dot{\varepsilon}_{SMALL}$	

References

[1] L. Lai, "Requirements documentation for engineering mechanics software: guidelines, template and a case study," M.A.Sc. Thesis, McMaster University, Jul. 2001.

[2] R. Ives. (2012) Bureau of Reclamation, U.S. Water Encyclopedia. [Online]. Available: http://www.waterencyclopedia.com/Bi-Ca/Bureau-of-Reclamation-U-S.html

[3] Hibernia. (2012) Photo Gallery. Hibernia. [Online]. Available: http://www.hibernia.ca/gallery/p13_platform.jpg

[4] L. Landau and E. Lifshitz, *Fluid Mechanics*, 2^{nd} ed. Toronto: Pergamon Press, 1987.

[5] D. Acheson, *Elementary Fluid Dynamics*. Oxford: Clarendon Press, 1990.

[6] M. Pringle, "Finite element modelling of two-component, solid-liquid mixtures," Ph.D. Thesis, McMaster University, Sep. 2004.

[7] R. Craig, *Craig's Soil Mechanics*, 7^{th} ed. New York, NY: Spon Press, 2004.

[8] J. Bear, *Dynamics of Fluids in Porous Media*. New York, NY: Dover Publications, 1972.

[9] D. Gidaspow, *Multiphase Flow and Fluidization, Continuum and Kinetic Theory Descriptions*. Boston, MA: Academic Press, 1994.

[10] S. Timoshenko and J. Goodier, *Theory of Elasticity*, 3rd ed. Toronto, ON: McGraw-Hill Publishing Company, Inc., 1970.

[11] EngineersToolbox. (2008) Hooke's Stress and Strain Calculation. EngineersToolbox. [Online]. Available: http://www.fea-optimization.com/ETBX/hooke_help.html

[12] B. Karchewski, "Module guide for two and three dimensional dynamic model of soil-water-structure interaction," Course Project - CES 741, McMaster University, Mar. 2012.

[13] ——, "Module interface specification for two and three dimensional dynamic model of soil-water-structure interaction," Course Project - CES 741, McMaster University, Mar. 2012.

[14] S. Kramer, *Geotechnical Earthquake Engineering*. Upper Saddle River, NJ: Prentice Hall, 1996.

[15] Plaxis bv. (2012) 2D Dynamics. Plaxis bv. [Online]. Available: http://www.plaxis.nl/shop/137/info//2D+Dynamics/

[16] ——. (2012) 3D Dynamics. Plaxis bv. [Online]. Available: http://www.plaxis.nl/shop/111/info//3D+Dynamics/

[17] OpenFOAM. (2011) OpenFOAM: The open source CFD toolbox. OpenFOAM. [Online]. Available: http://www.openfoam.com/

[18] ADINA. (2012) ADINA: Finite Element Analysis Software. ADINA. [Online]. Available: http://www.adina.com/index.shtml

[19] ANSYS. (2012) ANSYS: Simulation Driven Product Development. ANSYS. [Online]. Available: http://www.ansys.com/

[20] PLAXIS. (2012) PLAXIS: essential software for geotechnical professionals. PLAXIS. [Online]. Available: http://www.plaxis.nl/