

BridgeDesigner

A STRUCTURAL DESIGN AND OPTIMIZATION TOOLKIT

created by Brian Doran Giffin

September 8, 2024

CONTENTS

1	Introduction	3
2	Theoretical Development	5
2.1	Overview of Structural Finite Element Methods	5
2.1.1	Finite Element Equilibrium Equations	5
2.1.2	Assembly of Finite Element Stiffness Matrices	6
2.2	Structural Optimization	9
2.3	Objective	12
2.4	Section Property Derivatives	15
2.5	Element Geometry Derivatives	15
2.6	Element Stiffness Matrix	15
2.7	Element-wise Assembly Operations	17
3	Implementation in Matlab	18
4	User's Guide	19
4.1	Summary of MATLAB Classes in <code>BridgeDesigner</code>	19
4.1.1	Creating .dxf Files for Input to <code>BridgeDesigner</code>	20

INTRODUCTION

BridgeDesigner is a structural design and optimization toolkit that was created with the intent of aiding student teams in designing efficient structures for entry into the National Student Steel Bridge Competition. The primary goal of BridgeDesigner is the complete automation of the CAD-to-analysis work-flow, while maintaining a fairly simple and straightforward implementation in Matlab which can be easily modified to address the specific needs of the user.

Previous efforts towards improving the CAD-to-analysis process focused on three major tasks: the creation of analysis-ready models from existing CAD drawing files; the assessment of a given model's structural efficiency according to simulated results provided by the chosen analysis software; and the optimization of a given structural design (usually just the selection of optimal tubing section sizes for some, or all structural members.) The first tools aimed at automating these tasks (`proto.Bridge`, `fore.Score`, and `re.Struct`, respectively) were written in Visual Basic, provided Excel spreadsheet user interfaces, and relied upon the AutoCAD and SAP2000 application programming interfaces (APIs) to send and receive model data between the relevant programs. However, these tools suffered largely from platform-specific issues: either the software itself was unavailable (or not widely available), or the APIs had been deprecated for certain student versions of the programs – or, newer versions of the software employing different APIs produced run-time errors. All of these problems led to a great deal of effort on the part of users to get the design tools working on their personal computers, and for some, the tools were rendered inaccessible.

The BridgeDesigner structural design program aims to circumvent most of these issues, by avoiding the use of commercial APIs – for both AutoCAD and SAP2000 – altogether. Rather than retrieving drawing data from AutoCAD through API calls, BridgeDesigner is capable of processing files stored in the standardized `.dxf` drawing format, thereby allowing any version of AutoCAD (including student versions) to be used for automated design purposes. And, rather than interfacing with SAP2000, BridgeDesigner performs all of the relevant structural analysis computations internally using a simple and transparent structural finite element solver.

This document seeks to elaborate on the theoretical, implementational, and practical details regarding BridgeDesigner, in the hopes that it may see continued use, and possibly even open source development. Section 2 discusses some of the theoretical background behind the analysis methods employed by BridgeDesigner, while section 3 addresses the implementational layout of the finite element and optimization tools. Section 4 demon-

strates how to use the BridgeDesigner toolkit through Matlab scripts, and provides some relevant examples of how to use the available features for design purposes.

THEORETICAL DEVELOPMENT

This section summarizes some of the theoretical concepts underpinning the finite element method – a computational approach commonly used to solve large problems in structural analysis. This discussion is intended to aid and inform prospective users or developers about the corresponding capabilities (and limitations) of the BridgeDesigner analysis toolkit, and of the structural finite element method, more generally.

2.1 OVERVIEW OF STRUCTURAL FINITE ELEMENT METHODS

The finite element method as applied to structural engineering problems (sometimes also referred to as “matrix structural analysis” in this setting) is an approach which seeks to directly determine the deformations experienced by a given structure via the solution of a discrete system of linear algebraic equations. These equations characterize two primary requirements on a given structure: force/moment equilibrium, and compatibility. In a discrete setting, the resulting finite element system of equations relates two fundamental quantities of interest: joint forces/moments, and joint displacements/rotations.

2.1.1 Finite Element Equilibrium Equations

The generic finite element problem is succinctly expressed as a balance of forces:

$$\mathbf{f}^{\text{ext}} - \mathbf{f}^{\text{int}} = \mathbf{0}, \tag{1}$$

where \mathbf{f}^{ext} is a vector of all *externally applied* forces (and moments) acting on the joints of a structure, and \mathbf{f}^{int} is the corresponding vector of *internal* joint forces (and moments) induced by the internal stresses within all members comprising the structure. Henceforth, the term “force” (“displacement”) may be used to refer collectively to both forces and moments (displacements and rotations, respectively). For a structure to remain in equilibrium, the externally applied forces must balance with the internal forces, such that the *net* force acting on each joint is equal to zero.

If we assume that the material relationship remains linearly elastic ($\sigma = E\varepsilon$), and that the displacements of the joints are relatively small in relation to the size of the structure, then we may express the vector \mathbf{f}^{int} of internal forces as a direct linear function of the corresponding joint displacements \mathbf{u} via:

$$\mathbf{f}^{\text{int}} = \mathbf{K}\mathbf{u}. \tag{2}$$

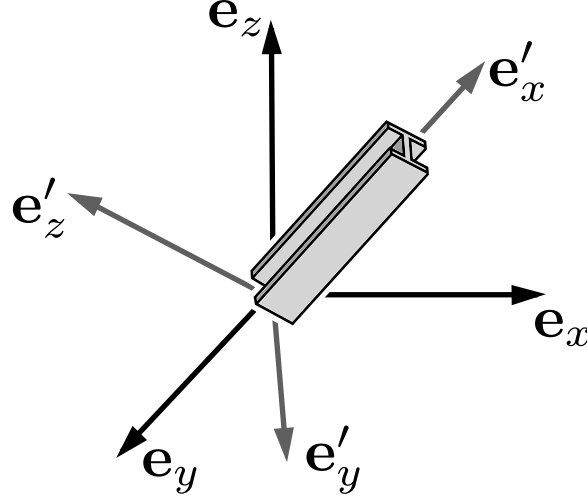


Figure 1: Depiction of the global (unprimed) and an element’s local (primed) coordinate systems.

The ensuing matrix \mathbf{K} is referred to as the “stiffness matrix” of the structure, which characterizes the (assumed linear) relationship between internal joint forces and joint displacements. Equation (2) may be regarded as a multi-dimensional generalization of Hooke’s law. Thus, for a given set of externally applied forces \mathbf{f}^{ext} , the associated joint displacements may be determined by inverting the stiffness matrix:

$$\mathbf{u} = \mathbf{K}^{-1} \mathbf{f}^{\text{ext}}. \quad (3)$$

Once the joint displacements have been obtained by solving the finite element system of equations (above), the internal forces/stresses within all members of the structure may be computed as a post-processing procedure.

2.1.2 Assembly of Finite Element Stiffness Matrices

Most of the work involved in solving a matrix structural analysis problem boils down to two main procedures: computing and assembling the stiffness matrix of the structure as a whole, and solving the resulting linear system of equations. The second step is a linear algebra problem, easily solved by the numerical packages available in Matlab. The only difficult task that remains is the assembly of the structure’s stiffness matrix.

To this end, it proves convenient to examine each member (or “element”) of a given structure independently. In doing so, one may construct an “element stiffness matrix” \mathbf{k}_e , which relates the vector of displacements \mathbf{u}_e applied to the ends of a single member, to the resulting joint forces $\mathbf{f}_e^{\text{int}}$ imposed by the element on the rest of the structure:

$$\mathbf{f}_e^{\text{int}} = \mathbf{k}_e \mathbf{u}_e. \quad (4)$$

In general, a given frame element may be oriented arbitrarily in 3-dimensional space, such that the axis of the member in question is not necessarily aligned with the “global” x -axis (refer to Figure 1.) For a variety of reasons, it proves convenient to define a “local coordinate system” for each element. Herein, quantities defined within an element’s local coordinate system are denoted by a primed index (e.g. \mathbf{u}'_e represents the vector of joint displacements oriented in the element’s local coordinate system).

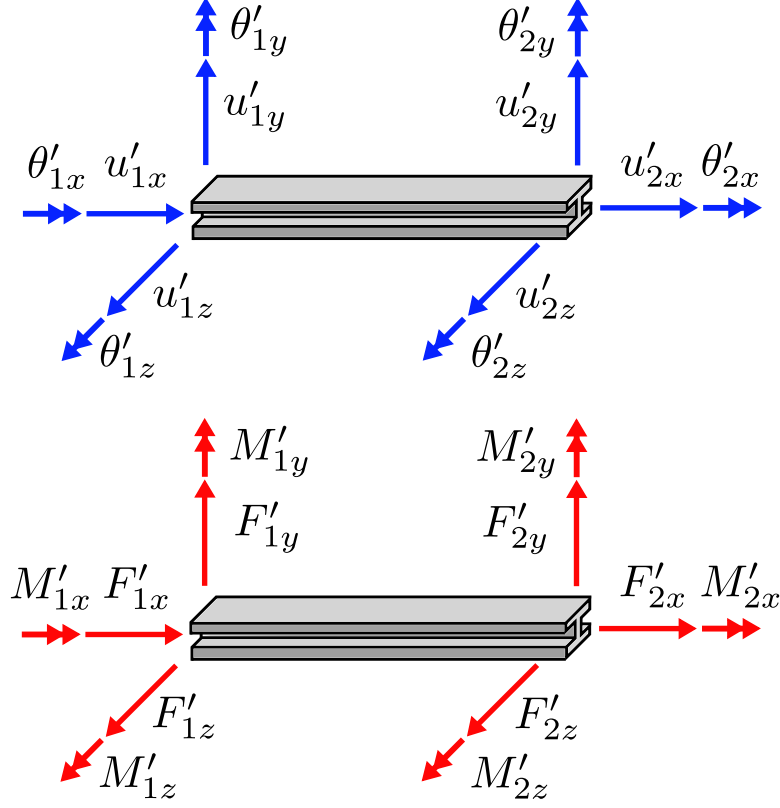


Figure 2: A representative 3D frame element oriented in its local coordinate system, with labeled joint displacements/rotations (top) and resulting internal joint forces/moments (bottom).

A traditional frame element possesses 2 joints: one at each of its end-points. In turn, each joint possesses 6 “degrees of freedom:” 3 translational displacements (u'_x , u'_y , u'_z) and 3 rotations (θ'_x , θ'_y , θ'_z). These displacements and rotations are “work-conjugated” with the corresponding internal forces (F'_x , F'_y , F'_z) and moments (M'_x , M'_y , M'_z) acting at a given joint.

The local stiffness matrix for a given frame element has dimensions of 12×12 , accounting for the 6 degrees of freedom at each of its 2 joints (refer to Figure 2.) When expressed in the element’s local coordinate system, the vectors of internal joint forces $\mathbf{f}_e^{\text{int}'}$ and joint displacements \mathbf{u}_e' are related via

$$\mathbf{f}_e^{\text{int}'} = \mathbf{k}_e' \mathbf{u}_e', \quad (5)$$

where

$$\mathbf{f}_e^{\text{int}'} \equiv \{ F'_{1x} \ F'_{1y} \ F'_{1z} \ M'_{1x} \ M'_{1y} \ M'_{1z} \ F'_{2x} \ F'_{2y} \ F'_{2z} \ M'_{2x} \ M'_{2y} \ M'_{2z} \}^T, \quad (6)$$

$$\mathbf{u}_e' \equiv \{ u'_{1x} \ u'_{1y} \ u'_{1z} \ \theta'_{1x} \ \theta'_{1y} \ \theta'_{1z} \ u'_{2x} \ u'_{2y} \ u'_{2z} \ \theta'_{2x} \ \theta'_{2y} \ \theta'_{2z} \}^T, \quad (7)$$

$$\mathbf{k}'_e \equiv \begin{bmatrix} \frac{AE}{L} & 0 & 0 & 0 & 0 & 0 & -\frac{AE}{L} & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{12EI_z}{L^3} & 0 & 0 & 0 & \frac{6EI_z}{L^2} & 0 & -\frac{12EI_z}{L^3} & 0 & 0 & 0 & \frac{6EI_z}{L^2} \\ 0 & 0 & \frac{12EI_y}{L^3} & 0 & -\frac{6EI_y}{L^2} & 0 & 0 & 0 & -\frac{12EI_y}{L^3} & 0 & -\frac{6EI_y}{L^2} & 0 \\ 0 & 0 & 0 & \frac{GJ}{L} & 0 & 0 & 0 & 0 & 0 & -\frac{GJ}{L} & 0 & 0 \\ 0 & 0 & -\frac{6EI_y}{L^2} & 0 & \frac{4EI_y}{L} & 0 & 0 & 0 & \frac{6EI_y}{L^2} & 0 & \frac{2EI_y}{L} & 0 \\ 0 & \frac{6EI_z}{L^2} & 0 & 0 & 0 & \frac{4EI_z}{L} & 0 & -\frac{6EI_z}{L^2} & 0 & 0 & 0 & \frac{2EI_z}{L} \\ -\frac{AE}{L} & 0 & 0 & 0 & 0 & 0 & \frac{AE}{L} & 0 & 0 & 0 & 0 & 0 \\ 0 & -\frac{12EI_z}{L^3} & 0 & 0 & 0 & -\frac{6EI_z}{L^2} & 0 & \frac{12EI_z}{L^3} & 0 & 0 & 0 & -\frac{6EI_z}{L^2} \\ 0 & 0 & -\frac{12EI_y}{L^3} & 0 & \frac{6EI_y}{L^2} & 0 & 0 & 0 & \frac{12EI_y}{L^3} & 0 & \frac{6EI_y}{L^2} & 0 \\ 0 & 0 & 0 & -\frac{GJ}{L} & 0 & 0 & 0 & 0 & 0 & \frac{GJ}{L} & 0 & 0 \\ 0 & 0 & -\frac{6EI_y}{L^2} & 0 & \frac{2EI_y}{L} & 0 & 0 & 0 & \frac{6EI_y}{L^2} & 0 & \frac{4EI_y}{L} & 0 \\ 0 & \frac{6EI_z}{L^2} & 0 & 0 & 0 & \frac{2EI_z}{L} & 0 & -\frac{6EI_z}{L^2} & 0 & 0 & 0 & \frac{4EI_z}{L} \end{bmatrix}. \quad (8)$$

The following quantities are herein defined, where the additional subscript e is used later on to distinguish between properties belonging to differing elements:

E : Young's modulus of the material

G : Shear modulus of the material ($G = \frac{E}{2(1+\nu)}$, with $\nu \equiv$ Poisson's ratio)

A : Area of the element's cross-section

I_y : Second moment of area about the element's local y' -axis

I_z : Second moment of area about the element's local z' -axis

J : Polar moment of area about the element's local x' -axis

L : Length of the element

All that remains is to transform the element's stiffness matrix into the global coordinate system. To this end, let \mathbf{Q} be defined as the 3×3 matrix which transforms the components of a vector \mathbf{v} represented in the global coordinate system, into a vector $\mathbf{v}' = \mathbf{Q}\mathbf{v}$ represented in the element's local coordinate system, i.e.

$$\mathbf{Q} = \begin{bmatrix} \mathbf{e}'_x \cdot \mathbf{e}_x & \mathbf{e}'_x \cdot \mathbf{e}_y & \mathbf{e}'_x \cdot \mathbf{e}_z \\ \mathbf{e}'_y \cdot \mathbf{e}_x & \mathbf{e}'_y \cdot \mathbf{e}_y & \mathbf{e}'_y \cdot \mathbf{e}_z \\ \mathbf{e}'_z \cdot \mathbf{e}_x & \mathbf{e}'_z \cdot \mathbf{e}_y & \mathbf{e}'_z \cdot \mathbf{e}_z \end{bmatrix}. \quad (9)$$

Further, let the following 12×12 "element transformation matrix" \mathbf{T} be defined as

$$\mathbf{T} = \begin{bmatrix} \mathbf{Q} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{Q} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{Q} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{Q} \end{bmatrix}. \quad (10)$$

It follows that

$$\mathbf{f}_e^{\text{int}} = \mathbf{T}^T \mathbf{f}_e^{\text{int}'}, \quad \mathbf{u}_e = \mathbf{T}^T \mathbf{u}_e', \quad \mathbf{k}_e = \mathbf{T}^T \mathbf{k}_e' \mathbf{T}. \quad (11)$$

2.2 STRUCTURAL OPTIMIZATION

Within the context of the Steel Bridge competition, our goal is to design the most efficient structure possible, as determined by a pre-defined “cost” function:

$$C(\Delta_{agg}, W) = f(\Delta_{agg}) + g(W), \quad (12)$$

where $C(\Delta_{agg}, W)$ is the structural efficiency cost function which depends on W (the total weight of the structure) and Δ_{agg} (the total measured aggregate deflection). Typically,

$$f(\Delta_{agg}) = c_\delta \Delta_{agg} \quad (13)$$

and

$$g(W) = c_w W \quad (14)$$

where c_w and c_δ are the associated costs per unit of weight and deflection, respectively.

The aggregate deflection $\Delta_{agg}(\mathbf{f})$ is a direct function of the applied loads \mathbf{f} acting on the structure. In the setting of the Steel Bridge competition, there are usually multiple possible loading cases \mathbf{f}_s where $s = 1, 2, \dots, N_{cases}$, which will therefore result in multiple possible measurements of the aggregate deflection $\Delta_{agg}(\mathbf{f}_s)$. For this reason, one must consider the probabilistic average aggregate deflection for use in the aforementioned cost function, i.e.

$$\Delta_{avg} = \sum_{s=1}^{N_{cases}} p(s) \Delta_{agg}(\mathbf{f}_s), \quad (15)$$

where $p(s)$ denotes the probability measure that load case s will occur during the competition. If all load cases have an equal chance of being selected, then we may consider the uniform probability distribution:

$$\Delta_{avg} = \frac{1}{N_{cases}} \sum_{s=1}^{N_{cases}} \Delta_{agg}(\mathbf{f}_s). \quad (16)$$

If we suppose that the joint displacements of the structure are obtained from $\mathbf{u}_s = \mathbf{K}^{-1}\mathbf{f}_s$ for a given load case s , then we may consider a “measurement vector” $\boldsymbol{\delta}_s$ which may be used to sample the deflections of specific joints in the structure (through a dot product with \mathbf{u}_s) to yield

$$\Delta_{agg}(\mathbf{f}_s) = \boldsymbol{\delta}_s \cdot \mathbf{u}_s = \boldsymbol{\delta}_s \cdot \mathbf{K}^{-1}\mathbf{f}_s. \quad (17)$$

It should be emphasized that both \mathbf{f}_s and $\boldsymbol{\delta}_s$ are fixed quantities which are determined by the competition scoring criteria regarding loading and deflection measurement locations.

In comparison, the weight of the structure W will remain unchanged regardless of which load case is selected, and may be computed via

$$W = \sum_{e=1}^{N_{members}} \rho A_e L_e \quad (18)$$

where ρ is the mass density of steel, and where A_e and L_e are the cross-sectional area and the length of a given member e .

For many years, both c_w and c_δ were taken to be constants which did not depend upon W or Δ_{agg} . With the advent of the 2018 NSSBC rules, c_δ remains a constant scoring

parameter, but the weight score $g(W)$ now depends implicitly upon the total weight W , such that

$$g(W) = \begin{cases} 0 & W < 120 \text{ lbs} \\ \$5,000(W - 120) & 120 \text{ lbs} \leq W < 200 \text{ lbs} \\ \$25,000(W - 184) & W > 200 \text{ lbs} \end{cases} \quad (19)$$

Alternatively, we may write

$$g(W) = \langle \$5,000(W - 120) \rangle + \langle \$20,000(W - 200) \rangle, \quad (20)$$

where $\langle x \rangle$ denotes the Macaulay bracket (or ramp) function, i.e. $\langle x \rangle = x$ if $x > 0$ and $\langle x \rangle = 0$ if $x \leq 0$.

The problem that we are faced with can be stated rather simply in mathematical terms:

$$\min_{\mathbf{a}} C(\Delta_{agg}(\mathbf{a}), W(\mathbf{a})) \quad (21)$$

where \mathbf{a} represents a vector of generic structural parameters of interest (e.g. the spatial locations of certain joints, or the cross-sectional properties of various members). Considering the latter, suppose we were to contemplate a fixed structural configuration. We should like to select optimal tubing sections for each member, whose cross-sectional areas $A_e \forall e \in 1, \dots, N_{members}$ will be taken as the minimizing parameters of interest. To solve for the optimal cross-sectional areas, we must first recognize that a minimum will be obtained when the derivative of the cost function with respect to each of the structural parameters is equal to zero:

$$J_e = \frac{\partial C}{\partial A_e} = 0 \quad \forall e \quad (22)$$

In mathematical terms, J_e is referred to as the ‘‘Jacobian’’ of the cost function C .

The above will yield a system of simultaneous equations in the variables A_e , which when solved will yield a first-order approximation to the optimal section sizes. Because the cost function is in general a non-linear function of the parameters A_e , successive Newton iteration is required to achieve convergence to the final optimum values of A_e .

Considering the first partial derivative of C with respect to the variables A_e , we obtain

$$\frac{\partial C}{\partial A_e} = \frac{\partial f}{\partial A_e} + \frac{\partial g}{\partial A_e}, \quad (23)$$

and

$$\frac{\partial f}{\partial A_e} = \frac{c_\delta}{N_{cases}} \sum_{s=1}^{N_{cases}} \boldsymbol{\delta}_s \cdot \frac{\partial \mathbf{K}^{-1}}{\partial A_e} \mathbf{f}_s, \quad \frac{\partial g}{\partial A_e} = c_w(W) \rho L_e, \quad (24)$$

where $c_w(W)$ is a piece-wise constant function of the total weight:

$$c_w(W) = \begin{cases} 0 & W < 120 \text{ lbs} \\ \$5,000 & 120 \text{ lbs} \leq W < 200 \text{ lbs} \\ \$25,000 & W > 200 \text{ lbs} \end{cases} \quad (25)$$

Using the chain rule of differentiation, we may write

$$\frac{\partial \mathbf{K}^{-1}}{\partial A_e} = \frac{\partial \mathbf{K}^{-1}}{\partial \mathbf{K}} \frac{\partial \mathbf{K}}{\partial A_e} = -\mathbf{K}^{-1} \mathbf{S}_e \mathbf{K}^{-1}, \quad (26)$$

where $\mathbf{S}_e = \partial \mathbf{K} / \partial A_e$ can be assembled from a single local element contribution \mathbf{s}_e :

$$\mathbf{s}_e = \begin{bmatrix} \frac{E}{L} & 0 & 0 & 0 & 0 & 0 & -\frac{E}{L} & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{12Er_g^2}{L^3} & 0 & 0 & 0 & \frac{6Er_g^2}{L^2} & 0 & -\frac{12Er_g^2}{L^3} & 0 & 0 & 0 & \frac{6Er_g^2}{L^2} \\ 0 & 0 & \frac{12Er_g^2}{L^3} & 0 & -\frac{6Er_g^2}{L^2} & 0 & 0 & 0 & -\frac{12Er_g^2}{L^3} & 0 & -\frac{6Er_g^2}{L^2} & 0 \\ 0 & 0 & 0 & \frac{2Gr_g^2}{L} & 0 & 0 & 0 & 0 & 0 & -\frac{2Gr_g^2}{L} & 0 & 0 \\ 0 & 0 & -\frac{6Er_g^2}{L^2} & 0 & \frac{4Er_g^2}{L} & 0 & 0 & 0 & \frac{6Er_g^2}{L^2} & 0 & \frac{2Er_g^2}{L} & 0 \\ 0 & \frac{6Er_g^2}{L^2} & 0 & 0 & 0 & \frac{4Er_g^2}{L} & 0 & -\frac{6Er_g^2}{L^2} & 0 & 0 & 0 & \frac{2Er_g^2}{L} \\ -\frac{E}{L} & 0 & 0 & 0 & 0 & 0 & \frac{E}{L} & 0 & 0 & 0 & 0 & 0 \\ 0 & -\frac{12Er_g^2}{L^3} & 0 & 0 & 0 & -\frac{6Er_g^2}{L^2} & 0 & \frac{12Er_g^2}{L^3} & 0 & 0 & 0 & -\frac{6Er_g^2}{L^2} \\ 0 & 0 & -\frac{12Er_g^2}{L^3} & 0 & \frac{6Er_g^2}{L^2} & 0 & 0 & 0 & \frac{12Er_g^2}{L^3} & 0 & \frac{6Er_g^2}{L^2} & 0 \\ 0 & 0 & 0 & -\frac{2Gr_g^2}{L} & 0 & 0 & 0 & 0 & 0 & \frac{2Gr_g^2}{L} & 0 & 0 \\ 0 & 0 & -\frac{6Er_g^2}{L^2} & 0 & \frac{2Er_g^2}{L} & 0 & 0 & 0 & \frac{6Er_g^2}{L^2} & 0 & \frac{4Er_g^2}{L} & 0 \\ 0 & \frac{6Er_g^2}{L^2} & 0 & 0 & 0 & \frac{2Er_g^2}{L} & 0 & -\frac{6Er_g^2}{L^2} & 0 & 0 & 0 & \frac{4Er_g^2}{L} \end{bmatrix}, \quad (27)$$

and where $r_g = \sqrt{I/A}$ represents the radius of gyration of the cross-section.

Ordinarily, we could pursue a solution to the above system of equations by taking the second derivative or ‘‘Hessian’’ of C , i.e.

$$H_{ek} = \frac{\partial^2 C}{\partial A_e \partial A_k} = \frac{\partial^2 f}{\partial A_e \partial A_k} + \frac{\partial^2 g}{\partial A_e \partial A_k}, \quad (28)$$

$$\frac{\partial^2 f}{\partial A_e \partial A_k} = \frac{c_\delta}{N_{cases}} \sum_{s=1}^{N_{cases}} \boldsymbol{\delta}_s \cdot \frac{\partial^2 \mathbf{K}^{-1}}{\partial A_e \partial A_k} \mathbf{f}_s, \quad \frac{\partial^2 g}{\partial A_e \partial A_k} = 0, \quad (29)$$

where

$$\frac{\partial^2 \mathbf{K}^{-1}}{\partial A_e \partial A_k} = \mathbf{K}^{-1} [\mathbf{S}_k \mathbf{K}^{-1} \mathbf{S}_e + \mathbf{S}_e \mathbf{K}^{-1} \mathbf{S}_k] \mathbf{K}^{-1}, \quad (30)$$

and

$$\frac{\partial^2 \Delta_{agg}(\mathbf{f}_s)}{\partial A_e \partial A_k} = \boldsymbol{\gamma}_s \cdot [\mathbf{S}_k \mathbf{K}^{-1} \mathbf{S}_e + \mathbf{S}_e \mathbf{K}^{-1} \mathbf{S}_k] \mathbf{u}_s = \boldsymbol{\Gamma}_{ks} \cdot \mathbf{K}^{-1} \mathbf{U}_{es} + \mathbf{U}_{ks} \cdot \mathbf{K}^{-1} \boldsymbol{\Gamma}_{es}, \quad (31)$$

We may then use the Hessian matrix to solve for the optimized cross-sectional areas within a Newton iteration procedure:

$$\mathbf{A}_{i+1} = \mathbf{A}_i - \mathbf{H}_i^{-1} \mathbf{J}_i. \quad (32)$$

There is a rather subtle issue at play here, however. Because of the inherent discontinuity in $c_w(W)$, we will most likely encounter convergence problems in our iteration procedure, particularly if the optimal weight of the structure lies at or near $W = 120$ or $W = 200$.

For this reason, we will assert the following postulate: *the optimal weight of the structure should lie in the range $120 \leq W \leq 200$* . This raises an interesting mathematical challenge, as we must incorporate a constraint into our optimization procedure. We will choose to do so through the use of a single Karush-Kuhn-Tucker constraint, yielding the Lagrangian of the cost function:

$$\Lambda(A_e, \mu) = C(A_e) + \mu(120 - W), \quad (33)$$

where $\mu \geq 0$ is a complementary slack variable, such that $\mu = 0$ when $(120 - W) < 0$ and $\mu > 0$ when $(120 - W) \geq 0$. In this setting, we may consider the Jacobian of the Lagrangian:

$$J_e = \frac{\partial \Lambda}{\partial A_e} = \frac{\partial C}{\partial A_e} - \mu \rho L_e = 0 \quad \forall e \quad (34)$$

$$J_\mu = \frac{\partial \Lambda}{\partial \mu} = 120 - \sum_{e=1}^{N_{members}} \rho A_e L_e = 0 \quad (35)$$

and the corresponding Hessian:

$$H_{ek} = \frac{\partial^2 \Lambda}{\partial A_e \partial A_k} = \frac{\partial^2 C}{\partial A_e \partial A_k}, \quad (36)$$

$$H_{e\mu} = \frac{\partial^2 \Lambda}{\partial A_e \partial \mu} = -\rho L_e. \quad (37)$$

The solution of the constrained minimum problem can be obtained from the saddle-point system of equations:

$$\begin{bmatrix} \mathbf{H}_{ek} & \mathbf{H}_{e\mu} \\ \mathbf{H}_{\mu k} & 0 \end{bmatrix} \begin{Bmatrix} \mathbf{A} \\ \mu \end{Bmatrix} = \begin{Bmatrix} \mathbf{J}_e \\ J_\mu \end{Bmatrix}. \quad (38)$$

In practice, this problem can be easily solved by first considering the unconstrained problem (assuming $\mu = 0$), checking whether the constraint $W > 120$ has been violated, and then optionally solving the above saddle-point problem to enforce the constraint (when $\mu > 0$).

The only problem that remains concerns the discrete nature of the available list of tubing sections. Consequently, it is recommended that the optimization be carried out in a continuous setting, and then mapped onto the discrete sub-space of available tubing section sizes via a closest-point projection method.

2.3 OBJECTIVE

Define the following quantities:

C : the total cost of the structure

Δ_s : the aggregate deflection of the structure for load case s

W : the total weight of the structure

c_Δ : the linear cost per unit of aggregate deflection

C_W : the total cost of the overall structure's weight (an implicit function of W)

\mathbf{f}_s : the net external loading on the structure for load case s

\mathbf{K} : the stiffness matrix for the structure

\mathbf{u}_s : the displacements of the structure for load case s

\mathbf{f}_s^* : the virtual unit dummy force vector for load case s

\mathbf{u}_s^* : the corresponding virtual displacement vector for load case s

$p(s)$: the probability of occurrence of load case s

ρ : the density of steel

E : the elastic modulus of steel

ν : the Poisson's ratio of steel

A_e : the cross-sectional area of member e

I_e : the moment of inertia of member e

J_e : the polar moment of inertia of member e

r_e : the radius of the section assigned to member e

t_e : the wall thickness of the section assigned to member e

L_e : the length of member e

\mathbf{a} : the vector of all structure variables to be optimized

Mathematically, the Steel Bridge optimization problem is defined as: *find the optimal values of all structural parameters \mathbf{a} such that the overall cost C of the structure is minimized, i.e.*

$$\min_{\mathbf{a}} C(\mathbf{a}) \quad (39)$$

where

$$C(\mathbf{a}) \equiv \sum_{s=1}^{N_s} p(s) C_{\Delta}(\Delta_s) + C_W(W), \quad (40)$$

$$\Delta_s(\mathbf{a}) \equiv \mathbf{f}_s^* \cdot \mathbf{K}^{-1} \cdot \mathbf{f}_s, \quad (41)$$

$$W(\mathbf{a}) \equiv \rho \sum_{e=1}^{N_e} A_e L_e. \quad (42)$$

The stationarity of $C(\mathbf{a})$ implies that an optimal set of parameter values has been chosen, i.e.

$$\mathbf{R}(\mathbf{a}) = \mathbf{0}, \quad (43)$$

where

$$\mathbf{R}(\mathbf{a}) \equiv \frac{\partial C}{\partial \mathbf{a}} = \sum_{s=1}^{N_s} p(s) \frac{\partial C_{\Delta}}{\partial \Delta} \frac{\partial \Delta_s}{\partial \mathbf{a}} + \frac{\partial C_W}{\partial W} \frac{\partial W}{\partial \mathbf{a}}, \quad (44)$$

$$\frac{\partial \Delta_s}{\partial \mathbf{a}} = \mathbf{f}_s^* \cdot \frac{\partial \mathbf{K}^{-1}}{\partial \mathbf{a}} \cdot \mathbf{f}_s, \quad (45)$$

$$\frac{\partial W}{\partial \mathbf{a}} = \rho \sum_{e=1}^{N_e} \left[\frac{\partial A_e}{\partial \mathbf{a}} L_e + A_e \frac{\partial L_e}{\partial \mathbf{a}} \right], \quad (46)$$

and

$$\frac{\partial \mathbf{K}^{-1}}{\partial \mathbf{a}} = -\frac{1}{2} \left[\mathbf{K}^{-1} \cdot \frac{\partial \mathbf{K}}{\partial \mathbf{a}} \cdot \mathbf{K}^{-1} + \left(\mathbf{K}^{-1} \cdot \frac{\partial \mathbf{K}}{\partial \mathbf{a}} \cdot \mathbf{K}^{-1} \right)^T \right]. \quad (47)$$

As a side note, it can be shown that any variation in the stiffness matrix $\delta \mathbf{K}$ due to a change in any structural parameter will always be symmetric, such that $\delta \mathbf{K} = (\delta \mathbf{K})^T$. This property permits the following simplification:

$$\frac{\partial \mathbf{K}^{-1}}{\partial \mathbf{a}} = -\mathbf{K}^{-1} \cdot \frac{\partial \mathbf{K}}{\partial \mathbf{a}} \cdot \mathbf{K}^{-1}, \quad (48)$$

$$\frac{\partial \Delta_s}{\partial \mathbf{a}} = -\mathbf{u}_s^* \cdot \frac{\partial \mathbf{K}}{\partial \mathbf{a}} \cdot \mathbf{u}_s. \quad (49)$$

A Newton-Raphson iterative procedure may be used to solve for the optimal values of \mathbf{a} . Specifically, while $\|\mathbf{R}(\mathbf{a}_k)\| > \epsilon$, compute an updated set of parameter values \mathbf{a}_{k+1} via

$$\mathbf{a}_{k+1} = \mathbf{a}_k - [\mathbf{J}(\mathbf{a}_k)]^{-1} \mathbf{R}(\mathbf{a}_k), \quad (50)$$

where

$$\mathbf{J}(\mathbf{a}) \equiv \frac{\partial^2 C}{\partial \mathbf{a} \otimes \partial \mathbf{a}}, \quad (51)$$

$$\frac{\partial^2 C}{\partial \mathbf{a} \otimes \partial \mathbf{a}} = \sum_{s=1}^{N_s} p(s) \frac{\partial^2 \Delta_s}{\partial \mathbf{a} \otimes \partial \mathbf{a}} + \frac{\partial C_W}{\partial W} \frac{\partial^2 W}{\partial \mathbf{a} \otimes \partial \mathbf{a}} + \frac{\partial^2 C_W}{\partial W^2} \left[\frac{\partial W}{\partial \mathbf{a}} \otimes \frac{\partial W}{\partial \mathbf{a}} \right], \quad (52)$$

$$\frac{\partial^2 \Delta_s}{\partial a_i \partial a_j} = \mathbf{f}_s^* \cdot \frac{\partial^2 \mathbf{K}^{-1}}{\partial a_i \partial a_j} \cdot \mathbf{f}_s, \quad (53)$$

$$\frac{\partial^2 W}{\partial \mathbf{a} \otimes \partial \mathbf{a}} = \rho \sum_{e=1}^{N_e} \left[\frac{\partial^2 A_e}{\partial \mathbf{a} \otimes \partial \mathbf{a}} L_e + \frac{\partial A_e}{\partial \mathbf{a}} \otimes \frac{\partial L_e}{\partial \mathbf{a}} + \frac{\partial L_e}{\partial \mathbf{a}} \otimes \frac{\partial A_e}{\partial \mathbf{a}} + A_e \frac{\partial^2 L_e}{\partial \mathbf{a} \otimes \partial \mathbf{a}} \right], \quad (54)$$

and

$$\frac{\partial^2 \mathbf{K}^{-1}}{\partial a_i \partial a_j} = \mathbf{K}^{-1} \cdot \left[\frac{\partial \mathbf{K}}{\partial a_i} \cdot \mathbf{K}^{-1} \cdot \frac{\partial \mathbf{K}}{\partial a_j} + \frac{\partial \mathbf{K}}{\partial a_j} \cdot \mathbf{K}^{-1} \cdot \frac{\partial \mathbf{K}}{\partial a_i} - \frac{\partial^2 \mathbf{K}}{\partial a_i \partial a_j} \right] \cdot \mathbf{K}^{-1}, \quad (55)$$

$$\frac{\partial^2 \Delta_s}{\partial a_i \partial a_j} = \mathbf{u}_s^* \cdot \left[\frac{\partial \mathbf{K}}{\partial a_i} \cdot \mathbf{K}^{-1} \cdot \frac{\partial \mathbf{K}}{\partial a_j} + \frac{\partial \mathbf{K}}{\partial a_j} \cdot \mathbf{K}^{-1} \cdot \frac{\partial \mathbf{K}}{\partial a_i} - \frac{\partial^2 \mathbf{K}}{\partial a_i \partial a_j} \right] \cdot \mathbf{u}_s. \quad (56)$$

Let the following terms be defined:

$$\left. \frac{\partial \mathbf{f}_s}{\partial a_i} \right|_{\mathbf{u}_s} = \frac{\partial \mathbf{K}}{\partial a_i} \cdot \mathbf{u}_s, \quad \left. \frac{\partial^2 \Delta_s}{\partial a_i \partial a_j} \right|_{\mathbf{u}_s, \mathbf{u}_s^*} = \mathbf{u}_s^* \cdot \frac{\partial^2 \mathbf{K}}{\partial a_i \partial a_j} \cdot \mathbf{u}_s. \quad (57)$$

Thus:

$$\frac{\partial^2 \Delta_s}{\partial a_i \partial a_j} = \left. \frac{\partial \mathbf{f}_s^*}{\partial a_i} \right|_{\mathbf{u}_s^*} \cdot \mathbf{K}^{-1} \cdot \left. \frac{\partial \mathbf{f}_s}{\partial a_j} \right|_{\mathbf{u}_s} + \left. \frac{\partial \mathbf{f}_s^*}{\partial a_j} \right|_{\mathbf{u}_s^*} \cdot \mathbf{K}^{-1} \cdot \left. \frac{\partial \mathbf{f}_s}{\partial a_i} \right|_{\mathbf{u}_s} - \left. \frac{\partial^2 \Delta_s}{\partial a_i \partial a_j} \right|_{\mathbf{u}_s, \mathbf{u}_s^*}. \quad (58)$$

In general, $\mathbf{J}(\mathbf{a})$ will be a dense matrix. Thus, for sufficiently large problems, it may be more efficient to move away from a gradient-based optimization method.

2.4 SECTION PROPERTY DERIVATIVES

Given a hollow circular section with outer radius r and wall thickness t , the section properties may be written as

$$A = \pi [r^2 - (r - t)^2], \quad I = \frac{\pi}{4} [r^4 - (r - t)^4], \quad I \equiv I_y = I_z = \frac{J}{2}. \quad (59)$$

The first and second (including mixed) partial derivatives of A and I with respect to r and t are:

$$\begin{bmatrix} A_{,r} & I_{,r} \\ A_{,t} & I_{,t} \end{bmatrix} = \begin{bmatrix} 2\pi t & \pi [r^3 - (r - t)^3] \\ 2\pi(r - t) & \pi(r - t)^3 \end{bmatrix}, \quad \begin{bmatrix} A_{,rr} & I_{,rr} \\ A_{,rt} & I_{,rt} \\ A_{,tt} & I_{,tt} \end{bmatrix} = \begin{bmatrix} 0 & 3\pi [r^2 - (r - t)^2] \\ 2\pi & 3\pi(r - t)^2 \\ -2\pi & -3\pi(r - t)^2 \end{bmatrix}. \quad (60)$$

2.5 ELEMENT GEOMETRY DERIVATIVES

$$\frac{\partial L_e}{\partial \mathbf{a}} = \boldsymbol{\lambda}_e \cdot \left[\frac{\partial \mathbf{X}_2^e}{\partial \mathbf{a}} - \frac{\partial \mathbf{X}_1^e}{\partial \mathbf{a}} \right], \quad \boldsymbol{\lambda}_e = \frac{\mathbf{X}_2^e - \mathbf{X}_1^e}{L_e}. \quad (61)$$

2.6 ELEMENT STIFFNESS MATRIX

$$\mathbf{f}_e = \{ F_{x1} \ F_{y1} \ F_{z1} \ M_{x1} \ M_{y1} \ M_{z1} \ F_{x2} \ F_{y2} \ F_{z2} \ M_{x2} \ M_{y2} \ M_{z2} \}^T, \quad (62)$$

$$\mathbf{u}_e = \{ u_{x1} \ u_{y1} \ u_{z1} \ \theta_{x1} \ \theta_{y1} \ \theta_{z1} \ u_{x2} \ u_{y2} \ u_{z2} \ \theta_{x2} \ \theta_{y2} \ \theta_{z2} \}^T, \quad (63)$$

$$\mathbf{f}_e = \mathbf{T}^T \mathbf{f}'_e, \quad (64)$$

$$\mathbf{u}'_e = \mathbf{T} \mathbf{u}_e, \quad (65)$$

$$\mathbf{k}_e = \mathbf{T}^T \mathbf{k}'_e \mathbf{T}, \quad (66)$$

$$\mathbf{T} = \begin{bmatrix} \mathbf{Q} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{Q} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{Q} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{Q} \end{bmatrix}, \quad (67)$$

$$\mathbf{Q} = \begin{bmatrix} \mathbf{e}'_1 \cdot \mathbf{e}_1 & \mathbf{e}'_1 \cdot \mathbf{e}_2 & \mathbf{e}'_1 \cdot \mathbf{e}_3 \\ \mathbf{e}'_2 \cdot \mathbf{e}_1 & \mathbf{e}'_2 \cdot \mathbf{e}_2 & \mathbf{e}'_2 \cdot \mathbf{e}_3 \\ \mathbf{e}'_3 \cdot \mathbf{e}_1 & \mathbf{e}'_3 \cdot \mathbf{e}_2 & \mathbf{e}'_3 \cdot \mathbf{e}_3 \end{bmatrix}, \quad (68)$$

$$\mathbf{k}'_e = \begin{bmatrix} k'_{1,1} & 0 & 0 & 0 & 0 & 0 & k'_{1,7} & 0 & 0 & 0 & 0 & 0 \\ 0 & k'_{2,2} & 0 & 0 & 0 & k'_{2,6} & 0 & k'_{2,8} & 0 & 0 & 0 & k'_{2,12} \\ 0 & 0 & k'_{3,3} & 0 & k'_{3,5} & 0 & 0 & 0 & k'_{3,9} & 0 & k'_{3,11} & 0 \\ 0 & 0 & 0 & k'_{4,4} & 0 & 0 & 0 & 0 & 0 & k'_{4,10} & 0 & 0 \\ 0 & 0 & k'_{5,3} & 0 & k'_{5,5} & 0 & 0 & 0 & k'_{5,9} & 0 & k'_{5,11} & 0 \\ 0 & k'_{6,2} & 0 & 0 & 0 & k'_{6,6} & 0 & k'_{6,8} & 0 & 0 & 0 & k'_{6,12} \\ k'_{7,1} & 0 & 0 & 0 & 0 & 0 & k'_{7,7} & 0 & 0 & 0 & 0 & 0 \\ 0 & k'_{8,2} & 0 & 0 & 0 & k'_{8,6} & 0 & k'_{8,8} & 0 & 0 & 0 & k'_{8,12} \\ 0 & 0 & k'_{9,3} & 0 & k'_{9,5} & 0 & 0 & 0 & k'_{9,9} & 0 & k'_{9,11} & 0 \\ 0 & 0 & 0 & k'_{10,4} & 0 & 0 & 0 & 0 & 0 & k'_{10,10} & 0 & 0 \\ 0 & 0 & k'_{11,3} & 0 & k'_{11,5} & 0 & 0 & 0 & k'_{11,9} & 0 & k'_{11,11} & 0 \\ 0 & k'_{12,2} & 0 & 0 & 0 & k'_{12,6} & 0 & k'_{12,8} & 0 & 0 & 0 & k'_{12,12} \end{bmatrix}, \quad (69)$$

$$\tilde{\mathbf{k}}_{u'_x}^{F'_x} = \frac{1}{A} \begin{bmatrix} k'_{1,1} & k'_{1,7} \\ k'_{7,1} & k'_{7,7} \end{bmatrix} = \frac{E}{L} \begin{bmatrix} +1 & -1 \\ -1 & +1 \end{bmatrix}, \quad (70)$$

$$\tilde{\mathbf{k}}_{\theta'_x}^{M'_x} = \frac{1}{J} \begin{bmatrix} k'_{4,4} & k'_{4,10} \\ k'_{10,4} & k'_{10,10} \end{bmatrix} = \frac{E}{2(1+\nu)L} \begin{bmatrix} +1 & -1 \\ -1 & +1 \end{bmatrix}, \quad (71)$$

$$\tilde{\mathbf{k}}_{u'_y}^{F'_y} = \frac{1}{I_z} \begin{bmatrix} k'_{2,2} & k'_{2,8} \\ k'_{8,2} & k'_{8,8} \end{bmatrix} = \frac{12E}{L^3} \begin{bmatrix} +1 & -1 \\ -1 & +1 \end{bmatrix}, \quad \tilde{\mathbf{k}}_{\theta'_z}^{F'_y} = \frac{1}{I_z} \begin{bmatrix} k'_{2,6} & k'_{2,12} \\ k'_{8,6} & k'_{8,12} \end{bmatrix} = \frac{6E}{L^2} \begin{bmatrix} +1 & +1 \\ -1 & -1 \end{bmatrix}, \quad (72)$$

$$\tilde{\mathbf{k}}_{\theta'_z}^{M'_z} = \frac{1}{I_z} \begin{bmatrix} k'_{6,6} & k'_{6,12} \\ k'_{12,6} & k'_{12,12} \end{bmatrix} = \frac{2E}{L} \begin{bmatrix} +2 & +1 \\ +1 & +2 \end{bmatrix}, \quad \tilde{\mathbf{k}}_{u'_y}^{M'_z} = \frac{1}{I_z} \begin{bmatrix} k'_{6,2} & k'_{6,8} \\ k'_{12,2} & k'_{12,8} \end{bmatrix} = \frac{6E}{L^2} \begin{bmatrix} +1 & -1 \\ +1 & -1 \end{bmatrix}, \quad (73)$$

$$\tilde{\mathbf{k}}_{u'_z}^{F'_z} = \frac{1}{I_y} \begin{bmatrix} k'_{3,3} & k'_{3,9} \\ k'_{9,3} & k'_{9,9} \end{bmatrix} = \frac{12E}{L^3} \begin{bmatrix} +1 & -1 \\ -1 & +1 \end{bmatrix}, \quad \tilde{\mathbf{k}}_{\theta'_y}^{F'_z} = \frac{1}{I_y} \begin{bmatrix} k'_{3,5} & k'_{3,11} \\ k'_{9,5} & k'_{9,11} \end{bmatrix} = \frac{6E}{L^2} \begin{bmatrix} -1 & -1 \\ +1 & +1 \end{bmatrix}, \quad (74)$$

$$\tilde{\mathbf{k}}_{\theta'_y}^{M'_y} = \frac{1}{I_y} \begin{bmatrix} k'_{5,5} & k'_{5,11} \\ k'_{11,5} & k'_{11,11} \end{bmatrix} = \frac{2E}{L} \begin{bmatrix} +2 & +1 \\ +1 & +2 \end{bmatrix}, \quad \tilde{\mathbf{k}}_{u'_z}^{M'_y} = \frac{1}{I_y} \begin{bmatrix} k'_{5,3} & k'_{5,9} \\ k'_{11,3} & k'_{11,9} \end{bmatrix} = \frac{6E}{L^2} \begin{bmatrix} -1 & +1 \\ -1 & +1 \end{bmatrix}, \quad (75)$$

$$\mathbf{f}_{F'_x} = \begin{Bmatrix} F'_{x1} \\ F'_{x2} \end{Bmatrix}, \quad \mathbf{u}_{u'_x} = \begin{Bmatrix} u'_{x1} \\ u'_{x2} \end{Bmatrix}, \quad \mathbf{f}_{F'_x} = A \left(\tilde{\mathbf{k}}_{u'_x}^{F'_x} \mathbf{u}_{u'_x} \right), \quad (76)$$

$$\mathbf{f}_{M'_x} = \begin{Bmatrix} M'_{x1} \\ M'_{x2} \end{Bmatrix}, \quad \mathbf{u}_{\theta'_x} = \begin{Bmatrix} \theta'_{x1} \\ \theta'_{x2} \end{Bmatrix}, \quad \mathbf{f}_{M'_x} = J \left(\tilde{\mathbf{k}}_{\theta'_x}^{M'_x} \mathbf{u}_{\theta'_x} \right), \quad (77)$$

$$\mathbf{f}_{F'_y} = \begin{Bmatrix} F'_{y1} \\ F'_{y2} \end{Bmatrix}, \quad \mathbf{u}_{u'_y} = \begin{Bmatrix} u'_{y1} \\ u'_{y2} \end{Bmatrix}, \quad \mathbf{f}_{M'_z} = \begin{Bmatrix} M'_{z1} \\ M'_{z2} \end{Bmatrix}, \quad \mathbf{u}_{\theta'_z} = \begin{Bmatrix} \theta'_{z1} \\ \theta'_{z2} \end{Bmatrix}, \quad (78)$$

$$\begin{Bmatrix} \mathbf{f}_{F'_y} \\ \mathbf{f}_{M'_z} \end{Bmatrix} = I_z \begin{bmatrix} \tilde{\mathbf{k}}_{u'_y}^{F'_y} & \tilde{\mathbf{k}}_{\theta'_z}^{F'_y} \\ \tilde{\mathbf{k}}_{u'_y}^{M'_z} & \tilde{\mathbf{k}}_{\theta'_z}^{M'_z} \end{bmatrix} \begin{Bmatrix} \mathbf{u}_{u'_y} \\ \mathbf{u}_{\theta'_z} \end{Bmatrix}, \quad (79)$$

$$\mathbf{f}_{F'_z} = \begin{Bmatrix} F'_{z1} \\ F'_{z2} \end{Bmatrix}, \quad \mathbf{u}_{u'_z} = \begin{Bmatrix} u'_{z1} \\ u'_{z2} \end{Bmatrix}, \quad \mathbf{f}_{M'_y} = \begin{Bmatrix} M'_{y1} \\ M'_{y2} \end{Bmatrix}, \quad \mathbf{u}_{\theta'_y} = \begin{Bmatrix} \theta'_{y1} \\ \theta'_{y2} \end{Bmatrix}, \quad (80)$$

$$\begin{Bmatrix} \mathbf{f}_{F'_z} \\ \mathbf{f}_{M'_y} \end{Bmatrix} = I_y \begin{bmatrix} \tilde{\mathbf{k}}_{u'_z}^{F'_z} & \tilde{\mathbf{k}}_{\theta'_y}^{F'_z} \\ \tilde{\mathbf{k}}_{u'_z}^{M'_y} & \tilde{\mathbf{k}}_{\theta'_y}^{M'_y} \end{bmatrix} \begin{Bmatrix} \mathbf{u}_{u'_z} \\ \mathbf{u}_{\theta'_y} \end{Bmatrix}, \quad (81)$$

$$\Delta_{,A}^e = \mathbf{u}_{u'_x}^* \cdot \tilde{\mathbf{k}}_{u'_x}^{F'_x} \mathbf{u}_{u'_x}, \quad \Delta_{,J}^e = \mathbf{u}_{\theta'_x}^* \cdot \tilde{\mathbf{k}}_{\theta'_x}^{M'_x} \mathbf{u}_{\theta'_x}, \quad (82)$$

$$\Delta_{,I_z}^e = \begin{Bmatrix} \mathbf{u}_{u'_y}^* & \mathbf{u}_{\theta'_z}^* \end{Bmatrix} \cdot \begin{bmatrix} \tilde{\mathbf{k}}_{u'_y}^{F'_y} & \tilde{\mathbf{k}}_{\theta'_z}^{F'_y} \\ \tilde{\mathbf{k}}_{u'_y}^{M'_z} & \tilde{\mathbf{k}}_{\theta'_z}^{M'_z} \end{bmatrix} \begin{Bmatrix} \mathbf{u}_{u'_y} \\ \mathbf{u}_{\theta'_z} \end{Bmatrix}, \quad \Delta_{,I_y}^e = \begin{Bmatrix} \mathbf{u}_{u'_z}^* & \mathbf{u}_{\theta'_y}^* \end{Bmatrix} \cdot \begin{bmatrix} \tilde{\mathbf{k}}_{u'_z}^{F'_z} & \tilde{\mathbf{k}}_{\theta'_y}^{F'_z} \\ \tilde{\mathbf{k}}_{u'_z}^{M'_y} & \tilde{\mathbf{k}}_{\theta'_y}^{M'_y} \end{bmatrix} \begin{Bmatrix} \mathbf{u}_{u'_z} \\ \mathbf{u}_{\theta'_y} \end{Bmatrix}, \quad (83)$$

$$\Delta^e = \mathbf{u}_e^* \cdot \mathbf{f}_e = \mathbf{u}_e'^* \cdot \mathbf{f}_e' = \Delta_{,A}^e A + \Delta_{,J}^e J + \Delta_{,I_z}^e I_z + \Delta_{,I_y}^e I_y. \quad (84)$$

For hollow circular (or square) cross-sections where $I \equiv I_y = I_z = \frac{J}{2}$:

$$\Delta^e = \Delta_{,A}^e A + \Delta_{,I}^e I, \quad \Delta_{,I}^e = 2\Delta_{,J}^e + \Delta_{,I_z}^e + \Delta_{,I_y}^e, \quad (85)$$

For hollow circular cross-sections, the first and second (including mixed) partial derivatives of Δ^e with respect to r and t are:

$$\begin{Bmatrix} \Delta_{,r}^e \\ \Delta_{,t}^e \end{Bmatrix} = \begin{bmatrix} A_{,r} & I_{,r} \\ A_{,t} & I_{,t} \end{bmatrix} \begin{Bmatrix} \Delta_{,A}^e \\ \Delta_{,I}^e \end{Bmatrix}, \quad \begin{Bmatrix} \Delta_{,rr}^e \\ \Delta_{,rt}^e \\ \Delta_{,tt}^e \end{Bmatrix} = \begin{bmatrix} A_{,rr} & I_{,rr} \\ A_{,rt} & I_{,rt} \\ A_{,tt} & I_{,tt} \end{bmatrix} \begin{Bmatrix} \Delta_{,A}^e \\ \Delta_{,I}^e \end{Bmatrix}. \quad (86)$$

The resulting mixed partial derivatives with respect to \mathbf{u}_e' and $\mathbf{u}_e'^*$ are:

$$\mathbf{f}_{e,A}'^* = \mathbf{k}_{,A}' \mathbf{u}_e'^*, \quad \mathbf{f}_{e,I}'^* = 2\mathbf{f}_{e,J}'^* + \mathbf{f}_{e,I_z}'^* + \mathbf{f}_{e,I_y}'^* = \left(2\mathbf{k}_{,J}' + \mathbf{k}_{,I_z}' + \mathbf{k}_{,I_y}' \right) \mathbf{u}_e'^*, \quad (87)$$

$$\mathbf{f}_{e,A}' = \mathbf{k}_{,A}' \mathbf{u}_e', \quad \mathbf{f}_{e,I}' = 2\mathbf{f}_{e,J}' + \mathbf{f}_{e,I_z}' + \mathbf{f}_{e,I_y}' = \left(2\mathbf{k}_{,J}' + \mathbf{k}_{,I_z}' + \mathbf{k}_{,I_y}' \right) \mathbf{u}_e', \quad (88)$$

$$\begin{Bmatrix} \mathbf{f}_{e,r}^* & \mathbf{f}_{e,t}^* \end{Bmatrix} = \mathbf{T}^T \begin{Bmatrix} \mathbf{f}_{e,A}'^* & \mathbf{f}_{e,I}'^* \end{Bmatrix} \begin{bmatrix} A_{,r} & A_{,t} \\ I_{,r} & I_{,t} \end{bmatrix}, \quad (89)$$

$$\begin{Bmatrix} \mathbf{f}_{e,r}' & \mathbf{f}_{e,t}' \end{Bmatrix} = \mathbf{T}^T \begin{Bmatrix} \mathbf{f}_{e,A}' & \mathbf{f}_{e,I}' \end{Bmatrix} \begin{bmatrix} A_{,r} & A_{,t} \\ I_{,r} & I_{,t} \end{bmatrix}, \quad (90)$$

2.7 ELEMENT-WISE ASSEMBLY OPERATIONS

A domain which has been discretized into finite elements may be exploited to the extent that many of the terms defined previously may be computed and assembled in an element-wise fashion. Considering a single element Ω_e , we may compute element-specific contributions of the form:

$$\left. \frac{\partial \mathbf{f}_s^e}{\partial a_i} \right|_{\mathbf{u}_s} = \frac{\partial \mathbf{k}^e}{\partial a_i} \cdot \mathbf{u}_s, \quad \frac{\partial \Delta_s^e}{\partial a_i} = -\mathbf{u}_s^* \cdot \frac{\partial \mathbf{k}^e}{\partial a_i} \cdot \mathbf{u}_s, \quad \left. \frac{\partial^2 \Delta_s^e}{\partial a_i \partial a_j} \right|_{\mathbf{u}_s, \mathbf{u}_s^*} = \mathbf{u}_s^* \cdot \frac{\partial^2 \mathbf{k}^e}{\partial a_i \partial a_j} \cdot \mathbf{u}_s. \quad (91)$$

— 3 —

IMPLEMENTATION IN MATLAB

This section is currently under construction...

USER'S GUIDE

This section briefly describes how to use the `BridgeDesigner` library in a practical design setting. `BridgeDesigner` is perhaps best thought of as an application programming interface (API) which may be incorporated into user-written MATLAB scripts. This enables designers to create their own analysis workflows, while handling most of the difficult/tedious design tasks under the hood. This guide provides an overview of the main API functions currently available for use in the `BridgeDesigner` library, and demonstrates their usage in a few simple example MATLAB scripts.

To use `BridgeDesigner`, you will need a working copy of Matlab (any relatively recent version will suffice). Open Matlab, and locate the folder named “BridgeDesigner” containing all of the program’s source files. You will need to add this folder to Matlab’s active “path” (right click and add to path, or simply enter the folder in the Matlab file explorer.)

4.1 SUMMARY OF MATLAB CLASSES IN `BRIDGEDESIGNER`

The `BridgeDesigner` library is comprised of a handful of MATLAB “class” definitions (see the [MathWorks documentation](#) pages for more details regarding classes.) These are:

Structure: An all-in-one structural analysis model, defining the problem geometry, external loading, support conditions, and cross-sectional properties of a given structure. This is the object of primary interest to a designer who wishes to set up, execute, and analyze a given structural analysis problem.

Joint: A single joint in a given frame structure.

Frame: A single frame member within a given structure with appropriately defined material and cross-sectional properties. A single frame member may be discretized into multiple finite elements for structural analysis purposes.

Element: A single finite element (belonging to a given frame member).

Material: A collection of material properties defining a given material type (i.e. steel).

Section: A collection of cross-sectional properties defining a given round or square hollow tubing section.

LoadCase: A combined description of the loads applied to a given structure (occurring over multiple loading steps), along with the deflection measurements to be taken throughout the loading process.

To define a Structure object in the **BridgeDesigner** framework, the geometry of the structure must be defined within a .dxf drawing file created in AutoCAD, along with additional meta-data necessary for interpreting the .dxf file and creating the analysis model.

4.1.1 Creating .dxf Files for Input to BridgeDesigner

The input format for the .dxf and accompanying meta-data must conform to a pre-specified set of standards. These standards are described herein.

Regarding the AutoCAD drawing file:

- The AutoCAD drawing file must be saved in .dxf format.
- The drawing must be dimensioned such that the primary units are inches.
- All frame elements should be drawn using the AutoCAD “line” command.
- Each line should correspond to a single continuous piece of tubing.
- All lines should be assigned to named layers in the AutoCAD drawing, with all lines belonging to the same layer sharing the same tubing size.
- All lines comprising the decking support surfaces of the structure must belong to the same layer, named as “Decking.”

Note: only the 2013 .dxf format has been thoroughly tested, but presumably the more recent formats should work, as well.

The meta-data comprises the following information:

- The coordinate directions in the AutoCAD drawing file corresponding to the “span” and “up” coordinate directions of the bridge (either: +x, -x, +y, -y, +z, or -z).
- The list of “active layers” that will be imported from the AutoCAD file, including: the name of each active layer to be imported, along with the name of its assigned tubing section size. Tubing names must be selected from the list of defined sections within the accompanying `defineProperties.m` script.