Selected Applications of the Theory of Connections: A Technique for Analytical Constraint Embedding

Hunter Johnston\textsuperscript{1,*}, Carl Leake\textsuperscript{1}, Yalchin Efendiev\textsuperscript{2}, Daniele Mortari\textsuperscript{1}

\textsuperscript{1}Department of Aerospace Engineering, Texas A&M University, College Station, TX 77843, USA
\textsuperscript{2}Department of Mathematics, Texas A&M University, College Station, TX 77843, USA

Abstract

In this paper, we consider several new applications of the recently introduced mathematical framework of the Theory of Connections (ToC). This framework transforms constrained problems into unconstrained problems by introducing constraint-free variables. Using this transformation, various ordinary differential equations (ODEs), partial differential equations (PDEs) and variational problems can be formulated where the constraints are always satisfied. The resulting equations can then be easily solved by introducing a global basis function set (e.g., Chebyshev, Legendre, etc.) and minimizing a residual at pre-defined collocation points. In this paper, we highlight the utility of ToC by introducing various problems that can be solved using this framework including: (1) analytical linear constraint optimization; (2) the brachistochrone problem; (3) over-constrained differential equations; (4) inequality constraints; and (5) triangular domains.

Keywords

linear constraint optimization; calculus of variation; over-constrained differential equations; inequality constraints; triangular domains; Theory of Connections

1. Introduction

The Theory of Connections (ToC), introduced by D. Mortari \cite{1} (see also \cite{2–4}), is a framework that transforms constrained problems into unconstrained problems by deriving “constrained expressions”. Using these expressions, the governing physical models are modified and solved in the space without constraints. Below, we present a brief overview. For univariate functions, Ref. \cite{1} introduced the constrained expression using the form,
\[ y(x) = g(x) + \sum_{k=1}^{n} \eta_k h_k(x), \]

where the \( h_k(x) \) are \( n \) assigned linearly independent functions, \( g(x) \) is a free function, and the coefficients \( \eta_k \) are derived by imposing the \( n \) constraints. The resulting constrained expression always satisfies the constraints, as long as \( g(x) \) is defined, where the constraints are defined.

As an example, we consider a function \( y(x) \) subject to the constraints \( y(x_1) = y_1 \) and \( y'(x_2) = y'_2 \). Following Equation (1), and selecting \( h_1(x) := 1 \) and \( h_2(x) := x \), the following two expressions are obtained [3,4].

\[
\begin{align*}
  y_1 &= g_1 + \eta_1 + \eta_2 x_1 \\
  y'_2 &= g'_2 + \eta_2
\end{align*}
\]

where \( g_1 \) and \( g'_2 \) are values of the free function and its derivatives evaluated at the two constraint points \( x_1 \) and \( x_2 \). The unknowns \( \eta_1 \) and \( \eta_2 \) can be solved leading to,

\[
\begin{align*}
  \eta_1 &= y_1 - g_1 - (y'_2 - g'_2)x_1 \\
  \eta_2 &= y'_2 - g'_2.
\end{align*}
\]

Plugging the \( \eta \) coefficients back into Equation (1) and simplifying, the final constrained expression is produced,

\[ y(x) = g(x) + (y_1 - g_1) + (y'_2 - g'_2)(x - x_1). \]

It can be seen that the constraints are always satisfied \( (y(x_1) = y_1 \) and \( y'(x_2) = y'_2) \), as long as \( g(x_1) \) and \( g'(x_2) \) are defined. This is an example that demonstrates Equation (1). The theory is general and can be used for many other constraints. In the paper, we will present other examples (see also [1,3] for more examples).

In general, the \( \eta_k \) coefficients are expressed in terms of the independent variables and in terms of \( g(x) \) evaluated where the constraints are defined. The Equation (1) is called constrained expression because it represents all possible functions satisfying the \( n \) constraints, regardless of the function \( g(x) \). Please note that \( g(x) \) can be discontinuous, partially defined, and even the Dirac delta function, as long as there are no constraints specified where the delta function is infinite.

The function \( y(x) \) in Equation (1) typically is applied to ODEs or some variational problems. In general, in the example of ODEs, we can consider a problem for \( y = y(x) \),

\[ F(x, y, y', ..., y^{(n)}) = 0, \]
subject to the constraints, which can include various types of constraints, (e.g., constraints on the function or derivatives, integral constraints, or so on). Here, \( F \) is some equation of \( y(x) \) and its derivatives. Using a ToC expression (Equation (1)), this equation is transformed to an unconstrained problem where the unknown parameter is defined by a new function \( g = g(x) \). This transformation can be written as,

\[
\bar{F}(x, g, g', ..., g^{(n)}) = 0,
\]

where \( \bar{F} \) is equations completely independent of the problem constraints. Moreover, \( \bar{F} \) differs from \( F \) and represents modified constitutive relations. The major advantage of Equation (2) is that it can be solved as an unconstrained optimization problem. Past work has focused on expanding the function of \( g(x) \) by a linear basis [5–7] such that

\[
g(x) = \xi^T h(x),
\]

where \( h(x) \) is a vector of global functions and \( \xi \) is vector of unknown coefficients used in the optimization process. These coefficients are determined by minimizing the residual of Equation (2) at some collocation points [8,9]. In general, one can use locally supported \( h(x) \) functions. One can also avoid a linear functional representation of \( g(x) \) and use a non-linear representation, for example, neural networks and support vector machines [10].

The Multivariate Theory of Connections [2] extends the original univariate theory [1] to \( n \)-dimensions and to any-degree boundary constraints. This extension can be summarized by the expression,

\[
y(x) = A(c(x)) + g(x) - A(g(x)),
\]

where \( x = \{x_1, x_1, ..., x_n\} \) is the vector of \( n \) orthogonal coordinates, \( c(x) \) is a function specifying the boundary constraints, \( A(c(x)) \) is any interpolating function satisfying the boundary constraints, and \( g(x) \) is the free function. Several examples of constrained expressions can be found in Refs. [1,2].

The major research effort of Theory of Connections has been applied to solve linear [4] and non-linear [3] ODEs. This has been done by expanding the free function, \( g(x) \), in terms of a set of basis functions (e.g., orthogonal polynomials, Fourier, etc.). Linear or iterative non-linear least-squares is then used to solve for the coefficients of this expansion. This approach to solve ODEs/PDEs has many advantages over traditional methods: (1) it consists of a unified framework to solve IVP, BVP, or multi-values problems, (2) it provides an approximated solution expressed via analytical functions that can be used for subsequent algebraic manipulation, (3) the solution accuracy is usually obtained fast for many application problems, (4) the procedure can be numerically robust (very low condition number), and (5) it can solve ODE/PDE subject to a variety of constraint types: absolute, relative, linear, non-linear, and integral. Additionally, this technique has recently been extended to solve 2-dimensional PDEs [10].
The purpose of this paper is to introduce a set of five new applications of Theory of Connections; applications that are not covered in the previous references and where ToC is found effective. The main motivation and a short rational description of the new applications considered in this paper are:

1. **Analytic linear constraint optimization.** In this application, the analytical problem of finding the extreme of a quadratic vectorial form subject to linear constraint is obtained using ToC instead of using the classical Lagrange multipliers technique.

2. **Brachistochrone problem.** This application is dedicated to one of the most famous problem in calculus of variations: the brachistochrone problem [11]. This problem is solved numerically and at machine error accuracy via the two-point boundary-value problem of the Euler-Lagrange equation associated with the brachistochrone functional integral definition. In particular, the solution is obtained by deriving the differential equation in polar coordinates.

3. **Over-constrained differential equations.** This application takes into consideration the problem of solving ODEs subject to more constraints than the degree of the ODE which arise in many areas of science and engineering. For example, in the problem of orbit determination of a satellite the number of measurements exceeds the order of the governing differential equation [12]. Furthermore, multi-purpose optimization [13–15] deals specifically with optimizing across multiple objective functions simultaneous where trade-offs (weighting) between two conflicting objectives must be taken into account. In this section, a weight least-squares solution is provided for over-constrained differential equations by assigning relative weights to the constraints and then solving the weighted constrained ODE by ToC. Additionally, another example showing the sequence of continuous solutions of an IVP morphing into an BVP is provided.

4. **Inequality constraints.** This application extends the ToC framework to include inequality constraints [16]. This is obtained using a combination of sigmoid functions to keep the constrained expression within the inequality constraints. This allows for the derivation of functions constrained by user-defined bounds than can be asymmetric, continuous, or symmetric discontinuous functions. The main motivation for this constraint type comes from optimal control problems (bounded control inputs).

5. **Triangular domains.** Validated by mathematical proof, Ref. [2] has extended the original univariate theory [1] to the multivariate theory [2]. This extension represents the multivariate formulation of ToC subject to arbitrary-order derivative constraints in rectangular domains. This provides an analytical procedure to obtain constrained expressions in any orthogonal/rectangular space that can be used to transform constrained problems into unconstrained problems. In Ref. [2] particular emphasis and details are given to the 2-dimensional case, because of the most important applications to surfaces (PDEs, Topography,
Visualization, etc.). This application begins the extension of ToC to triangular domains by deriving the surfaces satisfying boundary constraints.

The complete solution derivations of each one of these new applications, validated by numerical examples, are detailed in the subsequent sections.

2. Analytic Linear Constraints Optimization

This section shows an analytical application of the Theory of Connections. The problem, known as quadratic programming (QP) subject to equality constraints, consists of deriving a closed-form solution of the problem to find the min/max of the quadratic function, \( F(x) \), subject to \( m < n \) linear constraints,

\[
F(x) = \frac{1}{2} x^T Q x + c^T x \quad \text{subject to:} \quad Ax = b, \tag{3}
\]

where \( Q \in \mathbb{R}^{n \times n} \) and \( Q = Q^T > 0, c \in \mathbb{R}^n, A \in \mathbb{R}^{m \times n}, \) and \( b \in \mathbb{R}^m \), are all assigned, and \( D \) has rank \( m \) (all rows are independent). Let us search a solution in the form,

\[
x = g + A^T \eta.
\]

The constraint implies,

\[
A(g + A^T \eta) = b \quad \Rightarrow \quad AA^T \eta = b - Ag,
\]

and

\[
\eta = (AA^T)^{-1} b - (AA^T)^{-1} Ag.
\]

Therefore,

\[
x = g + A^T (AA^T)^{-1} b - A^T (AA^T)^{-1} Ag = x_0 + \left[ I_n - A^T (AA^T)^{-1} A \right] g = x_0 + Dg.
\]

Substituting this expression of \( x \) in Equation (3) we obtain,

\[
F(g) = \frac{1}{2} (x_0 + Dg)^T Q (x_0 + Dg) + c^T (x_0 + Dg).
\]

The problem of finding the extreme of Equation (3) is now an unconstrained optimization problem. Stationary condition implies

\[
\frac{df(g)}{dg} = 0 \quad \Rightarrow \quad Ag + d = 0 \tag{4}
\]

where
\[ \mathcal{A} = \frac{1}{2} D^T (Q + Q^T) D \quad \text{and} \quad d = \frac{1}{2} D^T (Q + Q^T) x_0 + D^T e. \]

In this example, we show how to transform a constrained optimization problem to unconstrained optimization problem and find the unconstrained \( g(x) \), which is the solution of Equation (4). This shows a simple application of Theory of Connections. Unfortunately, the matrix \( A \) is singular. However, Equation (4) is a consistent linear system. Therefore, the solution can be found in the null space of \( A \). This is computed by the Moore-Penrose inverse matrix (pseudo-inverse). Let \( A = C \Lambda C^T \) be the spectral decomposition of \( A \), where the eigenvector matrix, \( C \), is an orthogonal matrix because matrix \( A \) is symmetric. Therefore, the solution of the problem given in Equation (3) is,

\[ x = x_0 - D \Lambda^* C^T d, \]

where the diagonal matrix \( \Lambda^* \) contains the inverse of the nonzero eigenvalues and zero for zero eigenvalues.

3. Brachistochrone Problem

The brachistochrone problem is one of many problems arising from the calculus of variations. This body of work focuses on finding the extrema of functionals (mappings from functions to real numbers). Associated functions that maximize or minimize the functionals can be found using the Euler-Lagrange equation. In general, this equation produces an ODE which when solved produces the extremal function. The main application of ToC to the calculus of variation is solving the specific ODEs which can be highly complex.

Consider finding the function, \( f_s(x) \), minimizing the integral,

\[ I(f_s) = \min \int_a^b F(x, f(x), f'(x)) \, dx \tag{5} \]

where \( x \in [a, b] \), \( f(x) \) is an unknown function satisfying \( f(a) = f_a \) and \( f(b) = f_b \) and \( f'(b) = f_b \), and \( F(x, f(x), f'(x)) \) is the functional. An extremal, \( f_0(x) \), of the integral given in Equation (5) is the solution of the Euler-Lagrange equation,

\[ \frac{\partial F}{\partial f}(x, f_0(x), f_0'(x)) = \frac{d}{dx} \left[ \frac{\partial F}{\partial f'}(x, f_0(x), f_0'(x)) \right]. \tag{6} \]

The solution of this equation, \( f_0(x) \), is an extremal solution (where the derivative is zero, which is a necessary condition). This solution, however, is not sufficient as \( f_0(x) \) may be associated with some local minima, maxima, or saddle points. In other word, after solving Equation (6), we can consider \( f_s(x) = f_0(x) \), if the condition,
\[
\int_a^b F(x, f(x), f'(x)) \, dx \leq \min_{f(x)} \int_a^b F(x, f(x), f'(x)) \, dx, \quad \forall \ f(x)
\]

is verified. This can be done, for example, by evaluating the second derivative (Hessian) of \( F(x, f(x), f'(x)) \). Figure 1 shows three different coordinate systems that may be used to solve the brachistochrone problem.

Typically, the brachistochrone problem is written using coordinate system (a) of Figure 1. In this coordinate system, the total travel time of the sliding bead is given by the functional [11],

\[
F(x, y', y) = \sqrt{1 + y'^2} \frac{1}{2gy}, \quad y = y(x)
\]  

subject to,

\[
y(0) = 0 \quad \text{and} \quad y(x_1) = y_1.
\]

Using the steps of the calculus of variations outlined in Equation (6) on the function in Equation (7) yields the differential equation for the function \( y = y(x) \),

\[
2yy'' + y'^2 + 1 = 0.
\]

However, this coordinate system produces the case that \( y'(0) = \infty \) at the initial point and cannot be solved using ToC. If the axes are oriented as shown in coordinate system (b) of Figure 1, the functional used in the calculus of variations becomes,

\[
F(x, y', y) = \sqrt{1 + y'^2} \frac{1}{2gx}, \quad y = y(x)
\]

which after performing the steps of the calculus of variations leads to the differential equation for the function \( y = y(x) \),

\[
2xy'' + y'^3 + y' = 0.
\]

The problem with the differential equation given in Equation (9) is that the solution may take on two different values of \( y(x) \) for a given value of \( x \). Analytically, the issues with the DEs given in Equations (8) and (9) are circumvented by solving the problems parametrically. Unfortunately, the DE for the brachistochrone written in the parametric form either contains not-a-number or infinite values when implemented numerically, or admits a trivial solution. For example, let \( x(p) \) and \( y(p) \) each be functions of the parameter \( p \) and their derivatives with respect to \( p \) be given by the notation: \( dx/dp = x_p \) and \( d^2x/dp^2 = x_{pp} \). Then, Equation (9) can be rewritten using the chain rule as
\[
2x_y^{yy} \frac{y^{pp}}{x^{pp}} + \left( \frac{y_p}{x_p} \right)^3 + \frac{y_p}{x_p} = 0, \quad x = x(p), \quad y = y(p).
\] (10)

Unfortunately, Equation (10) will have infinite values anytime \(x_p = 0\) or \(x_{pp} = 0\). Therefore, it cannot be implemented numerically. Alternatively, one could write the DE as
\[
2x y_p^3 + x_{pp} \left( y_p x_p^2 + y_p^3 \right) = 0, \quad x = x(p), \quad y = y(p).
\] (11)

However, Equation (11) admits the trivial solution \(x = 0\). Attempts have been made to avoid the trivial solution by giving a starting value for the non-linear least-squares that is far from the solution \(x = 0\), but none of these attempts were successful.

Another option is to express the brachistochrone problem in polar coordinates (coordinate system (c) in Figure 1). If the coordinates are chosen correctly, then the problems of the Cartesian coordinate systems can be avoided, and the problem does not need to be parameterized. A downside of this approach is a significantly more complicated DE; however, by using the ToC method, solving this DE is very easy. Using this coordinate system, the energy of an object sliding without friction is given by,
\[
\frac{1}{2} m v^2 = m g r \sin \theta, \quad r = r(\theta).
\] (12)

In addition, the differential path length in polar coordinates is given by,
\[
ds = \sqrt{r^2 + r'^2} \, d\theta,
\] (13)
where \(r' = dr/d\theta\). Furthermore, using Equation (12), the velocity, \(v\), can be written as,
\[
v = \frac{ds}{dt} = \sqrt{2gr \cos \theta}.
\] (14)

Using Equations (13) and (14) one can express the total time to get from the start point to the end point as
\[
T = \int_0^t \, dt = \int_{\theta_0}^{\theta_f} \frac{\sqrt{r^2 + r'^2}}{\sqrt{2gr \cos \theta}} \, d\theta.
\] (15)

Let the integrand of Equation (15) be defined as the functional \(F(\theta, r(\theta), r'(\theta))\). Then, the calculus of variations can be used to minimize the travel time. This leads to the DE, after some simplification, given in Equation (16).
\[
-r^2(\dot{r} \tan \theta + 2\ddot{r}) - \tan \theta \dot{r}^3 + 3\dot{r} r + r^3 = 0, \quad r = r(\theta).
\] (16)
Using the differential equation given by Equation (16), the ToC method was used to solve
the particular case with initial condition \((x_0, y_0) = (0, 0)\) and final condition \((x_f, y_f) = (1, 5)\).
Figure 2 shows information associated with solving the brachistochrone problem and
highlights the error between the ToC solution and the real solution. Additionally, the
residuals of the DE are between the order of \(10^{-12}\) and \(10^{-16}\) when the ToC formulation is
used and \(g(x)\) is expressed by 60 Chebyshev polynomials. It can be seen that this method
can estimate the solution of the brachistochrone problem with high accuracy.

4. Over-Constrained Differential Equations

The seminal paper on ToC \([1]\) presented Equation (1) as a way to incorporate \(n\) constraints
to form a constrained expression, this equation is repeated below:

\[
y(x) = g(x) + \sum_{k=1}^{n} \eta_k h_k(x).
\]

This process leads to a \((n \times n)\) matrix populated by the values of \(h_k(x)\) evaluated at the
constraint conditions. The values of \(\eta_k\) can then be solve by simply inverting this matrix.
Yet, through this process, it is not required that number of \(\eta\) coefficients equals the number
of applied constraints, and thus constraint matrix can also be written for \(n\) values of \(\eta\) and \(m\)
constraints. Which leads to,

\[
M \eta = \begin{bmatrix}
h_1^{(d_1)}(x_1) & h_2^{(d_1)}(x_1) & \cdots & h_n^{(d_1)}(x_1) \\
h_1^{(d_2)}(x_2) & h_2^{(d_2)}(x_2) & \cdots & h_n^{(d_2)}(x_2) \\
\vdots & \vdots & \ddots & \vdots \\
h_1^{(d_m)}(x_m) & h_2^{(d_m)}(x_m) & \cdots & h_n^{(d_m)}(x_m)
\end{bmatrix}
\begin{bmatrix}
\eta_1 \\
\eta_2 \\
\vdots \\
\eta_n
\end{bmatrix}
= \begin{bmatrix}
y_1^{(d_1)} - g_1^{(d_1)} \\
y_2^{(d_2)} - g_2^{(d_2)} \\
\vdots \\
y_m^{(d_m)} - g_m^{(d_m)}
\end{bmatrix}.
\]

where \(d_i\) denotes the order of derivative of the constraint at point \(x_i\) where \(i \in [1, m]\). This
becomes an over-determined system with the constraint matrix \(M\) being \((m \times n)\). The \(\eta\)
values can still be solved by incorporating a weighted least-squares technique, where we
define \(W\) as a diagonal weighting matrix,

\[
W = \begin{bmatrix}
w_1 & 0 & \cdots & 0 \\
0 & w_2 & \cdots & 0 \\
\vdots & \vdots & \ddots & \vdots \\
0 & \cdots & 0 & w_m
\end{bmatrix}.
\]

leading to the solution,
\[ \eta = (M^T W M)^{-1} M^T W \begin{bmatrix} (d_1) - g_1 \\ (d_2) - g_2 \\ \vdots \\ (d_m) - g_m \end{bmatrix}. \]

Unlike the traditional ToC approach where the \( \eta \) values produced an equation always satisfying the given constraints, the weight least-squares approach produces a weighted constrained expression which satisfies the constraints in a weighted (relative) sense. To exactly solve a Differential Equation, the number of constraints MUST be equal to the order of the Differential Equation. Yet, through the ToC method, the constrained expression is formed and then the differential equation is solved. This separation allows for the incorporation of more constraints than the order of the Differential Equation. The concept of this is highlighted in Figure 3 which provides an outline which distinguishes the ToC approach from classical methods in interpolation and least-squares. Part (c) in Figure 3 shows the original formulation of ToC where the number of the \( \eta \) coefficients was equal to the number of constraints incorporated. Part (d) highlights the work of this section which combines this general interpolation method with a weighted least-squares technique for the constraints.

4.1. Second Order Differential Equation with Three Point Constraints. A Numerical Example

As a numerical test, let us consider solving the Differential Equation,

\[ y'' + 2y' + y = 0, \quad y = y(x). \quad (17) \]

For this example, let us assume that a dynamical system described by Equation (17) is “observed” at three points \( x = [-1, -0.5, +1] \) and these measurements are subject to normally distributed noise such that

\[ y(-1) = N(y_1^{\text{true}}, \sigma_1^2), \quad y(-0.5) = N(y_2^{\text{true}}, \sigma_2^2), \quad \text{and} \quad N(y_3^{\text{true}}, \sigma_3^2), \]

where for this problem \( y_1^{\text{true}} = 5, y_2^{\text{true}} = 4.515, \) and \( y_3^{\text{true}} = 2. \) For this, we define the weight matrix based on the variances,

\[ W := \begin{bmatrix} w_1 & 0 & 0 \\ 0 & w_2 & 0 \\ 0 & 0 & w_3 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & \sigma_2^{-2} & 0 \\ 0 & 0 & \sigma_3^{-2} \end{bmatrix}. \]

Using the development in the prior section, this Differential Equation can incorporate information from all three observations even though the Differential Equation is only
second-order. Since the Differential Equation is second-order, the solution must be searched by the following constrained expression,

\[ y(x) = g(x) + \eta_1 b_1(x) + \eta_2 b_2(x). \]

Then the values of \( \eta_1 \) and \( \eta_2 \) are computed by weighted least-squares where we define

\[ b_1(x) := 1 \quad \text{and} \quad b_2(x) := x, \]

\[ \eta_1 \eta_2 = \left( M^T W M \right)^{-1} M^T W \begin{bmatrix} y_1 - g_1 \\ y_2 - g_2 \\ y_3 - g_3 \end{bmatrix}, \quad M := \begin{bmatrix} 1 & x_1 \\ 1 & x_2 \\ 1 & x_3 \end{bmatrix} \]

where the \( M \) matrix is defined by imposing the constraints and the \( W \) matrix is populated by the scaled variances above. The solution leads to a constrained expression of the form,

\[ y(x) = g(x) + \beta_1(x)(y_1 - g_1) + \beta_2(x)(y_2 - g_2) + \beta_3(x)(y_3 - g_3), \quad (18) \]

such that \( \Delta_{ij} := x_i - x_j \). Equation (18) represents all functions satisfying constraints relative to the given variances \( \sigma_1^2, \sigma_2^2, \text{and} \sigma_3^2 \). With the constraints completely captured in Equation (18), the \( g(x) \) function represents the solution space that satisfies the three constraints by least-squares. Now let us move toward solving Equation (17). First, let the free function \( g(x) \), be expressed as a linear combination of a given basis set where \( g(x) = \xi^T h(x) \).

Substituting this into Equation (18) leads to,

\[ y(x) = \xi^T h(x) + \beta_1(x)(y_1 - \xi^T h_1) + \beta_2(x)(y_2 - \xi^T h_2) + \beta_3(x)(y_3 - \xi^T h_3), \]

where the constrained expression can be reduced to the form,

\[ y(x) = \xi^T a(x) + b(x), \]

This form now has transformed the constrained optimization problem into an unconstrained optimization problem on \( \xi \), and can be solved using the techniques developed in Ref. [3,4].

In this specific problem, a Monte Carlo simulation of 10,000 trials was conducted to determine space that the function \( y(x) \) could occupy given the “observation” uncertainty and subject to dynamics governed by the Differential Equation. Figure 4 highlights this solution.

In this figure, plot (a) shows the Differential Equation solution space given the observation uncertainty and plot (b) highlights the residuals of the Differential Equation over the entire
simulation. It can be seen that the residuals of all solutions are between $10^{-13}$ and $10^{-14}$. Plots (c), (d), and (e) display the distribution of the constraint points around the true value. Note: these values are sampled from the solutions of the Differential Equation and not the constraints specified in the constrained expression.

Through this test, a probability bound for the Differential Equation can be produced, along with an estimated mean. For all solutions, the residual of the differential remained less than $10^{-13}$ verifying the accuracy of the method. Additionally, an interesting result of this test is in the final estimated solutions. Most evident in the constraints at $x = -0.5$ and $x = +1$, the $3\sigma$ of the Differential Equation is less than that of the observation $3\sigma$. This arises because the loss function in the ToC method minimizes the residuals of the Differential Equation, and now in the over-constrained ToC method the residuals are minimized simultaneously with the weighted least-squares of the observations (or constraints).

### 4.2. Initial to Boundary-Value Problem Transformation. A Numerical Example

Another application made possible by this extension is a continuous transformation of an initial value problem to a boundary-value problem. The following numerically validates this approach. Consider solving,

$$y'' + [\cos(3x^2) - 3x + 1]y' + \left[6\sin(4x^2) - e^{\cos(3x)}\right]y = 2 \frac{[1 - \sin(3x)](3x - \pi)}{4 - x}, \quad y = y(x),$$

subject to the three constraints

$$y_0 = y(-1) = -2, \quad y'_0 = y'(-1) = -2, \quad \text{and} \quad y_f = y(+1) = 2.$$

Since the Differential Equation is second-order, the constrained expression takes the form,

$$y(x) = g(x) + \eta_1 h_1(x) + \eta_2 h_2(x).$$

Since the highest order constraint is the first derivative, we can safely define $h_1(x) := 1$ and $h_2(x) := x$. Then the values of $\eta_1$ and $\eta_2$ are computed again by weighted least-squares,

$$\begin{bmatrix} \eta_1 \\ \eta_2 \end{bmatrix} = \left(M^T W M\right)^{-1} M^T W \begin{bmatrix} y_0 - g_0 \\ y'_0 - g'_0 \\ y_f - g_f \end{bmatrix},$$

where the $M$ matrix is defined by imposing the constraints and the $W$ matrix is a diagonal matrix of weights, i.e.,

$$M := \begin{bmatrix} 1 & -1 \\ 0 & 1 \\ 1 & 1 \end{bmatrix} \quad \text{and} \quad W := \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 - \alpha & 0 \\ 0 & 0 & \alpha \end{bmatrix}.$$
and $\alpha$ is a weight parameter such that $\alpha \in [0, 1]$, which is synonymous with the transformation from the IVP to the BVP. From this derivation, $V$ becomes

$$V = \frac{1}{1 + 4\alpha - a^2} \begin{bmatrix} a + 1 & (1 - a)^2 & 1 + 2a - a^2 \\ -2a & 1 - a^2 & 2a \end{bmatrix}.$$

and solving for $\eta$ leads again Equation (18) of the form,

$$y(x) = g(x) + \beta_1(x)(y_1 - g_1) + \beta_2(x)(y_2 - g_2) + \beta_3(x)(y_3 - g_3),$$

where the $\beta$ terms are defined by,

$$\beta_1(x) = \frac{1 + a[1 - 2x]}{1 + 4a - a^2},$$

$$\beta_2(x) = \frac{(1 - a)^2 + [1 - a^2]}{(1 + 4a - a^2)},$$

$$\beta_3(x) = \frac{1 + 2a[1 + x] - a^2}{1 + 4a - a^2}.$$

Again, $g(x)$ can be defined as $g = \xi^T h(x)$ and the unknown coefficient vector $\xi$ can be solved with least-squares. Figure 5 shows this transformation “surface” along with the residuals of the Differential Equation for validation of the method. It can be seen that the mean residual overall $\alpha$ values are on the order of $10^{-14}$ with a standard deviation on the same order. This plot shows the solution of the Differential Equation, $y(x)$, continuously morphing from initial point and derivative constraints to initial and final point constraints.

5. Inequality Constraints

In some cases, it may be desired to find all possible trajectories between two functions, $f_u(x)$ and $f_d(x)$, that represent upper and lower continuous bounds, respectively. This means that $f_u(x) > f_d(x)$, $\forall \ x \in [x_{min}, x_{max}]$. All trajectories between these two functions can be approximated by the following expression,

$$y(x) = g(x) + [f(x) - g(x)]\sigma(k, g(x) - f_d(x)) + [f_d(x) - g(x)]\sigma(k, f_d(x) - g(x)) \quad (19)$$

where $g(x)$ represents the free function. In the cases that there are also equality constraints to be applied, the function $g(x)$ can be replaced by Equation (1). Additionally,

$$\sigma(k, z) = \frac{1}{1 + e^{-kz}} \text{ and } \frac{d\sigma(k, z)}{dz} = k\sigma(k, z)(1 - \sigma(k, z)).$$

is a sigmoid function with smoothing parameter $k$. The sigmoid function acts as a “switching function” that subtracts off the portion of $g(x)$ that exceeds the bounds. In this equation, $k$ is not infinite, and thus $y(x)$ will experience some overshooting/undershooting when $g(x) > f_u(x)$ or $g(x) < f_d(x)$ that must be quantified. To analyze this, let us consider a simple case where the function is only constrained from above with $f_u(x)$ and therefore only overshooting exists.
\[ y(x) = g(x) + \left[ f_u(x) - g(x) \right] \sigma(k, g(x) - f_u(x)). \]  

(20)

For this case, the overshooting phenomenon is shown in Figure 6.

Since the only free part of the equation is \( g(x) \), the maximum value of the entire function, \( y(x) \), will be when the derivative \( \frac{dy(x)}{dg(x)} = 0 \). Performing this derivative, Equation (20) becomes,

\[ \frac{dy}{dg} = 0 = 1 - \frac{e^{-k(g(x) - f_u(x))}k(f_u(x) - g(x))}{1 + e^{-k(g(x) - f_u(x))}^2}. \]

which can be solved for \( g(x) \). The resulting value, \( g(x)_{\text{max}} \), is the value of \( g(x) \) that will cause the maximum overshoot in \( y(x) \). This process yields,

\[ g(x)_{\text{max}} = \frac{kf_u(x) + \text{productlog}(e^{-1})}{k} + 1 \]

where productlog is the product logarithm function (also-called the omega function or the Lambert W function). Substituting this result back into Equation (19), we can obtain the maximum value that \( y(x) \) can take.

\[ y(x)_{\text{max}} = f_u(x) + \frac{\text{productlog}(e^{-1})}{k}. \]  

(21)

Now, we use Equation (21) to solve for a modified upper bound, \( \hat{f}_u(x) \), such that the maximum value of \( y(x) \) is \( f_u(x) \); \( \hat{f}_u(x) \) is the adjusted boundary shown in Figure 6. This yields Equation (22).

\[ \hat{f}_u(x) = f_u(x) - \frac{\text{productlog}(e^{-1})}{k}. \]  

(22)

A similar derivation can be made for the lower bound \( \hat{f}_d(x) \); the solution is shown in Equation (23).

\[ \hat{f}_d(x) = f_d(x) + \frac{\text{productlog}(e^{-1})}{k}. \]  

(23)

The functions \( \hat{f}_u(x) \) and \( \hat{f}_d(x) \) replace \( f_u(x) \) and \( f_d(x) \) in Equation (19), resulting in,

\[ y(x) = g(x) + \left[ f_u(x) - g(x) \right] \sigma(k, g(x) - f_u(x)) \]

\[ + \left[ f_d(x) - g(x) \right] \sigma(k, f_d(x) - g(x)). \]  

(24)
Equation (21) shows that the value of $k$ is the only user-selected parameter that affects the difference between $y(x)_{\max}$ and $f_u(x)$, because $\text{productlog}(e^{-1})$ is a constant. Increasing the value of $k$ will decrease the difference between $y(x)_{\max}$ and $f_u(x)$, and also the difference between $y(x)_{\min}$ and $f_d(x)$. Decreasing the value of $k$ will have the opposite effect. Equation (24) was used to generate Figure 7, which contains randomly generated inequality boundaries and functions. This figure shows that all the random functions stay within the randomly generated inequality boundaries, and thus demonstrates how this technique can be used to enforce inequality boundary constraints.

6. Triangular Domains

This section demonstrates how to construct a constrained expression with a free function, $g(x, y)$, that will always meet the Dirichlet boundary conditions on a triangular domain. This section is broken down into two subsections. The first proves that an affine transformation exists that can transform the unit triangle into any desired triangle: shown in Figure 8. The second subsection shows how to build the constrained expression for the unit triangle, and how to extend it, via the affine transformation given in the first subsection, to any triangular domain.

6.1. Affine Transformation from the Unit Triangle to the Generic Triangle

Consider a unit triangle, which is shown on the left plot of Figure 8. This unit triangle has the vertices defined by the vectors, $P_1 := (0, 0)^T$, $P_2 := (0, 1)^T$, and $P_3 := (1, 0)^T$.

This subsection presents an affine transformation, denoted by $A$, that transforms the coordinates of a unit triangle defined in the $[X,Y]$ (capital frame) into the coordinates $[x, y, z]$ of the triangle in the desired coordinate frame.

Let the three vectors $P_i$, where $i \in \{1, 2, 3\}$, be the position vectors of vertices of the unit triangle in the starred frame. Similarly, $p_i$, where $i \in \{1, 2, 3\}$, be the position vectors of the vertices of the desired triangle in the regular frame. Then, the affine coordinate transformation given in Equation (25) transforms the unit triangle in the capital frame into the desired triangle in the desired frame.

$$
\begin{bmatrix}
  x \\
  y \\
  z \\
\end{bmatrix} = \begin{bmatrix} p_3 - p_1 \\ p_2 - p_1 \\ (p_3 - p_1) \times (p_2 - p_1) \end{bmatrix} \begin{bmatrix}
  X \\
  Y \\
  0 \\
\end{bmatrix} + p_1 = A \begin{bmatrix}
  X \\
  Y \\
  0 \\
\end{bmatrix} + p_1. 
$$

(25)

The following identities prove the mapping provided by Equation (25),

$$
p_1 = A \begin{bmatrix}
  0 \\
  0 \\
  0 \\
\end{bmatrix} + p_1, \quad p_2 = A \begin{bmatrix}
  0 \\
  1 \\
  0 \\
\end{bmatrix} + p_1, \quad \text{and} \quad p_3 = A \begin{bmatrix}
  1 \\
  0 \\
  0 \\
\end{bmatrix} + p_1.
$$

The inverse transformation is also of interest, and can be written as,
\[
\begin{pmatrix}
X \\
Y
\end{pmatrix} = \begin{pmatrix}
p_3 - p_1, & p_2 - p_1, & (p_3 - p_1) \times (p_2 - p_1)
\end{pmatrix}^{-1} \begin{pmatrix}
x \\
y \\
z
\end{pmatrix} - p_1.
\tag{26}
\]

The inverse transformation will always exist if the triangle is non-degenerate. The first column of \(A\) is the vector that points from \(p_3\) to \(p_1\), and the second column is the vector from \(p_2\) to \(p_1\). For a triangle, these two vectors will never be parallel. The third column of \(A\) is defined as the cross-product of the first two columns, and will therefore be orthogonal to the vectors that make up the first two columns. Therefore, the columns of \(A\) are linearly independent, and \(A\) is invertible. Therefore, the inverse transform always exists.

### 6.2. Coons-Type Surface on the Unit Triangle

Consider unit triangle with boundary constraints provided by the boundary functions, 
\(F_1(0, Y), F_2(X, 1 - X) = F_2(1 - Y, Y)\) when \(Y + X = 1\), and \(F_3(X, 0)\). For this unit triangle, the following Coons-type surface can be constructed,

\[
F(X, Y) = (1 - X - Y)[F_3(X, 0) + F_1(0, Y) - F_3(0, 0)] + \\
+X[F_3(X, 0) - F_3(1 - Y, 0) + F_2(1 - Y, Y)] + \\
+Y[F_1(0, Y) - F_1(0, 1 - X) + F_2(X, 1 - X)].
\tag{27}
\]

From Equation (27) one can show that

\[
F(X, 0) = F_3(X, 0), \quad F(0, Y) = F_1(0, Y), \quad \text{and} \quad \begin{cases}
F(X, 1 - X) = F_2(X, 1 - X) \\
F(1 - Y, Y) = F_2(1 - Y, Y).
\end{cases}
\]

Let’s write Equation (27) in the following compact form,

\[
F(X, Y) = C(F_1, F_2, F_3)
\tag{28}
\]

where \(C(F_1, F_2, F_3)\) is a function of the boundary constraints, \(F_1(0, Y), F_2(X, 1 - X) = F_2(1 - Y, Y), \) and \(F_3(X, 0)\).

### 6.3. ToC Surfaces on the Unit Triangle

Equation (28) can be specified for any surface, \(Z = G(X, Y)\). This means that we can write the identity,

\[
G(X, Y) = C(G_1, G_2, G_3)
\]

where \(G_1 = G(0, Y), G_2 = G(X, 1 - X)\) or \(G_2 = G(1 - Y, Y), \) and \(G_3 = G(X, 0)\). Therefore, ToC allows us to write the constrained expression representing all surfaces satisfying the boundary conditions on the unit triangle as,
\[ F(X, Y) = C(F_1, F_2, F_3) + G(X, Y) - C(G_1, G_2, G_3) \]  

(29)

where \( G(X, Y) \) is a free function. The constrained expression given in Equation (29) can be written explicitly as,

\[
F(X, Y) = G(X, Y) + (1 - X - Y) [F_3(X, 0) - G(X, 0) + F_1(0, Y) - G(0, Y) - F_1(0, 0) + G(0, 0)] + \
X[F_3(X, 0) - G(X, 0) - F_3(1 - Y, 0) + G(1 - Y, 0) + F_2(1 - Y, Y) - G(1 - Y, Y)] + \
Y[F_1(0, Y) - G(0, Y) - F_1(0, 1 - X) + G(0, 1 - X) + F_2(X, 1 - X) - G(X, 1 - X)].
\]

### 6.4. ToC Surfaces on the Generic Triangle

The ToC surfaces on the generic triangle can be derived using Equation (29) and the affine mapping given in Equation (26). Setting, \( X = (X, Y) \), Equation (29) can be rewritten as

\[
F(X) = C[F_1(X), F_2(X), F_3(X)] + G(X) - C[G_1(X), G_2(X), G_3(X)].
\]

Setting, \( x = (x, y, z) \), the affine transformation can be written as

\[
X = A^{-1} (x - p_1),
\]

and the ToC Surfaces on the Generic Triangle can be written as,

\[
F(x) = C[F_1((X(x)), F_2((X(x)), F_3((X(x)))) + G((X(x))) - C[G_1((X(x)), G_2((X(x)), G_3((X(x))))]],
\]

\[
F_1 = (6 - y)(y - 1.5)\cos(3x) + 4, \\
F_2 = (2 - x)(x - 3)\exp(x)\cos(2y) + 4, \\
F_3 = (y - 1.5)(x - 3)\sqrt{xy} + 4,
\]

and the vertices of the triangle were \( P_1 = (1, 1.5), P_2 = (2, 6), \) and \( P_3 = (3, 1) \).

Figures 9 and 10 show that the ToC surfaces meet the triangular boundary constraints (shown in black) regardless of the free function, \( g(x, y) \), chosen.

### 7. Conclusions

This work demonstrated some of the applications of ToC as a unified framework. The first application showed how to solve QP problems subject to equality constraints without the use of Lagrange multipliers. Using ToC, the equality constraints are embedded into a constrained expression that transforms the problem into an unconstrained optimization problem. The second application showed how to solve one of the most famous calculus of variations problems, the brachistochrone problem, with ToC. The calculus of variations technique was used to derive a highly non-linear differential equation in polar coordinates, to avoid singularities, which was solved via ToC. The third application explored solving over-constrained differential equations, where there are more constraints than the order of the
differential equation. Using ToC through weighted least-squares, we transform the problem into unconstrained problem, which can easily be solved. The fourth application showed how to extend ToC to incorporate inequality constraints by using sigmoid functions. In addition, a discussion on avoiding undershoot/overshoot was included. The fifth and final application showed how to write constrained expressions for triangular domains subject to Dirichlet boundary conditions.

Future work will investigate how some of these applications can be extended to solve optimal control problems. The QP problem subject to linear equality constraints and solving the brachistochrone problem represent the first research thrust in that direction, as these are some of the classic problems in the field of optimal control. Moreover, future work will investigate how these applications can be combined with the application on inequality constraints to solve optimal control problems subject to inequality constraints.

Furthermore, future work will investigate applications of over-constrained differential equations. One example is applications in orbit determination where measurements exceed the order of the differential equation governing the dynamics of the orbiting body. By using the approach developed for weighted least-squares solutions of differential equations, the dynamics of the body and variance of observational measurements can be combined into a single optimization problem. Another area of future work includes multi-objective optimization problems.

Additionally, future research will look to extend constrained expressions on triangular domains to include constraints on the derivatives normal to the sides of the triangle. The obvious application of this research extension will be in solving differential equations on triangular domains; however, this research extension also has applications in meshing, domain discretization, and visualization.

Acknowledgments

Funding: This work was supported by the NASA Space Technology Research Fellowship (NSTRF): Hunter Johnston (NSTRF 2019), Carl Leake (NSTRF 2019)

Abbreviations

The following abbreviations are used in this manuscript:

- **BVP** boundary-value problem
- **DE** differential equation
- **IVP** initial value problem
- **ODE** ordinary differential equation
- **PDE** partial differential equation
- **QP** quadratic programming
- **ToC** Theory of Connections
References

1. Mortari D The Theory of Connections: Connecting Points. Mathematics 2017, 5, 57.
2. Mortari D; Leake C The Multivariate Theory of Connections. Mathematics 2019, 7, 296.
3. Mortari D; Johnston H; Smith L High accuracy least-squares solutions of nonlinear differential equations. J. Comput. Appl. Math. 2019, 352, 293–307. [PubMed: 32454554]
4. Mortari D Least-squares Solution of Linear Differential Equations. Mathematics 2017, 5, 48.
5. Gottlieb D; Orszag SA Numerical Analysis of Spectral Methods: Theory and Applications; Society for Industrial and Applied Mathematics: Philadelphia, PA, USA, 1977.
6. Jain MK Numerical Solution of Differential Equations; Wiley Eastern: New Delhi, India, 1979.
7. Driscoll TA; Hale N Rectangular spectral collocation. IMA J. Numer. Anal. 2016, 36, 108–132.
8. Lanczos C Applied Analysis In Progress in Industrial Mathematics at ECMI 2008; Dover Publications, Inc: New York, NY, USA, 1957; Chapter 7, p. 504.
9. Wright K Chebyshev Collocation Methods for Ordinary Differential Equations. Comput. J 1964, 6, 358–365.
10. Leake C; Johnston H; Smith L; Mortari D Approximating Ordinary and Partial Differential Equations Using the Theory of Connections and Least Squares Support Vector Machines. Trans. Neural Netw. Learn. Syst 2019, submitted.
11. Hurtado JE Paul Bunyan’s Brachistochrone and Tautochrone. J. Astronaut. Sci. 2000, 48, 207–224.
12. Bate RR; Mueller DD; White JE Fundamentals of Astrodynamics; Dover Publications, Inc.: New York, NY, USA, 1971; Chapter 2.
13. Miettinen K Nonlinear Multiobjective Optimization In International Series in Operations Research & Management Science; Springer: New York, NY, USA, 1999.
14. Shirazi A; Najaf B; Aminyavari M; Rinaldi F; Taylor RA Thermal economic environmental analysis and multi-objective optimization of an ice thermal energy storage system for gas turbine cycle inlet air cooling. Energy 2014, 69, 212–226.
15. Ellefsen K; Lepikson H; Albiez J Multiobjective coverage path planning: Enabling automated inspection of complex, real-world structures. Appl. Soft Comput. 2017, 61, 264–282.
16. Lewis FL; Vrabie DL; Syrmos VL Optimal Control; John Wiley & Sons, Inc.: Hoboken, NJ, USA, 2012.
Figure 1.
Brachistochrone Coordinate Systems: (a) classic cartesian coordinate system, (b) modified cartesian coordinate system, and (c) polar coordinate system.
Figure 2.
Polar brachistochrone differential equation solution error and residuals.
Figure 3.
The classical formulation of (a) interpolation and (b) least-squares provides a single function solution. The ToC method provides the numerical framework to describe (c) all possible functions satisfying the constraints and (d) all possible functions of a least-squares fitting.
Figure 4.
Monte Carlo test for 10,000 trials. Part (a) represents the solutions of the differential equation over the varying observed “constraints” with part (b) quantifying the accuracy of the solution. Parts (c)–(e) represent the distribution of the solution values compared with the true value of the “constraints”.
Figure 5.
IVP to BVP Differential Equation parametric transformation.
Figure 6.
Overshoot of $y(x)$ for the upper bound constraint. This phenomenon can be quantified and can be used to eliminate the overshooting.
Figure 7.
Random constrained expressions with continuous bounds.
Figure 8.
Affine Transformation from Unit Triangle to Desired Triangle.
Figure 9.
ToC Triangular Surface with $g(x, y) = 0$. 
Figure 10.
ToC Triangular Surface with \( g(x, y) = \cos\left(\frac{1}{2}\exp x\right)\sin(5y)\sqrt{x} \).