Differential Programming via OR Methods

Shannon Sweitzer
Department of Industrial and Systems Engineering, University of Southern California

T. K. Satish Kumar
Department of Computer Science, Department of Physics and Astronomy, Department of Industrial and Systems Engineering, Information Sciences Institute, University of Southern California

Abstract

Systems of ordinary differential equations (ODEs) and partial differential equations (PDEs) are extensively used in many fields of science, including physics, biochemistry, nonlinear control, and dynamical systems. On the one hand, analytical methods for solving systems of ODEs/PDEs mostly remain an art and are largely insufficient for complex systems. On the other hand, numerical approximation methods do not yield a viable analytical form of the solution that is often required for downstream tasks. In this paper, we present an approximate approach for solving systems of ODEs/PDEs analytically using solvers like Gurobi developed in Operations Research (OR). Our main idea is to represent entire functions as Bézier curves/surfaces with to-be-determined control points. The ODEs/PDEs as well as their boundary conditions can then be reformulated as constraints on these control points. In many cases, this reformulation yields quadratic programming problems (QPPs) that can be solved in polynomial time. It also allows us to reason about inequalities. We demonstrate the success of our approach on several interesting classes of ODEs/PDEs.

2012 ACM Subject Classification Applied Computing

Keywords and phrases Differential Programming, Operations Research, Bézier Curves.

Digital Object Identifier 10.4230/LIPIcs.CP.2021.45

1 Introduction

Systems of ordinary differential equations (ODEs) and partial differential equations (PDEs) are so extensively used that they are the mathematical language of many sciences. The following are just a few examples. In physics, they are used to describe harmonic motion, radioactive decay, and propagation of electromagnetic waves [18]. In biochemistry, they are used to model biological processes ranging from the biosynthesis of phospholipids and proteins to the growth of cancer cells and viral dynamics [2]. In control theory, they are used to describe the behavior of dynamical systems and optimal strategies for controlling them [6]. An abundance of other applications can be found in many other sciences.

Plenty of analytical methods have been developed for solving ODEs/PDEs. These include standard techniques like separation of variables, the method of characteristics, integral transform, change of variables, fundamental solutions, and superposition [14], and newer techniques that utilize Lie groups [13] and Bäcklund transforms [9]. Despite the existence of many such techniques, the applicability of analytical methods has been restricted to special classes of systems of ODEs/PDEs such as first-order systems, second-order systems with constant coefficients, and second-order systems with variable coefficients. A comprehensive study of ODEs/PDEs amenable to different analytical methods can be found in [14].

Many numerical approximation methods have also been developed to address the limitations of analytical methods. These include the finite element method (FEM) [20], the finite difference method (FDM) [10], and the finite volume method (FVM) [3]. Although numerical approximation methods have very general applicability, they too have drawbacks of their own. They don't return a solution in an analytical form. Instead, they return a

solution as a set of values calculated at discrete points on a meshed geometry. This makes the solution unviable for downstream tasks in which analytical operations may be involved.

While fitting high-order polynomials on the set of values calculated at discrete points can sometimes salvage an analytical form and enable some downstream derivatives, it is not a satisfactory method either. This is because enforcing a desired property, such as an inequality constraint, at the discrete points does not enforce it everywhere. For example, in a system of ODEs that describes the velocity profile of a robot, enforcing a maximum velocity constraint on a finite number of discrete time points doesn't enforce it at all time points if high-order polynomials are used for interpolation.[1]

Because of the problems associated with both analytical and numerical approximation methods, controller design and synthesis problems in many domains still remain very hard. Such problems involve decision variables in addition to ODEs/PDEs. Typically, the ODEs/PDEs can be solved in a "simulation" phase only when decision variables have assigned values. Neither the analytical nor the numerical approximation method can facilitate the search for optimal values of the decision variables themselves. For example, in nanoscale photonics, a set of teflon dielectric cylinders are used to focus electromagnetic power. Given values for the decision variables, i.e., positions of the dielectric cylinders, the PDEs that describe the resulting distribution of electromagnetic power can be solved numerically. However, the optical filter design problem of choosing where to optimally place the dielectric cylinders themselves is very hard.

In this paper, we present an approximate approach for solving systems of ODEs/PDEs analytically using solvers like Gurobi developed in Operations Research (OR). Our main idea is to represent entire functions as Bézier curves/surfaces with to-be-determined control points. Bézier curves have a number of useful mathematical properties [11]. They can uniformly approximate any continuous function; their derivatives are also Bézier curves; and a Bézier curve lies entirely within the convex hull of its control points. Because of their attractive mathematical properties, Bézier curves have been widely used in many application domains, including computer graphics [12], computer-aided design [5], path planning [1], and trajectory planning [16]. Using the Bézier curve/surface representation in our case, we show that ODEs/PDEs as well as their boundary conditions can be reformulated as constraints on their control points.

Our proposed approach has several advantages. First, not only does it have the general applicability of numerical approximation methods but it also produces an analytical form that is useful for downstream tasks. Such downstream tasks are commonplace in physics-based machine learning where the *principle of least action* can be expressed as a second-order PDE, known as the Euler-Lagrange equation [15], on which further data-driven inferences must be carried out.

Second, our approach uses *control points* instead of a *discretization* of the independent variables' domains. For example, consider a function $f(t)$ with $t \in [0, T]$. Numerical approximation methods require the discretization of the interval $[0, T]$. However, discretization not only necessitates an increase in the number of discrete points for growing values of T but also creates a dependency on interpolation methods for values of t between the discrete points. In contrast, our approach represents $f(t)$ as a linear combination of Bernstein basis

[1] There exist other numerical approximation methods, called meshfree methods [7], useful for simulations in which the discrete points can be dynamically created or destroyed. Meshfree methods can also be combined with FEM, FDM, or FVM to yield hybrid methods [4]. But, in general, they too have the same drawbacks.

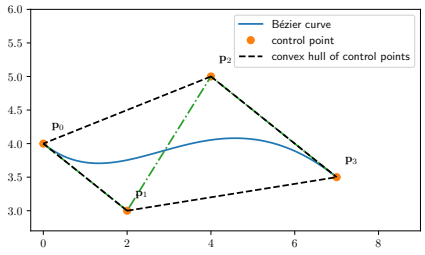


Figure 1 Illustrates an important property of Bézier curves: A Bézier curve is enclosed entirely within the convex hull of its control points. Here, the Bézier curve has 4 control points in a 2-dimensional space.

polynomials on t. The to-be-determined coefficients of the linear combination are its control points. $f(t)$ is therefore automatically defined for all values of $t \in [0, T]$. Moreover, the number of control points depends on the complexity of $f(t)$ and not on the domain size of t. In fact, some of the benefits of such a representation are used in recent numerical approximation methods like isogeometric analysis [8].

Third, although our approach uses only a finite number of control points, it allows us to enforce desired properties at all points on the resulting solution rather than at a set of discrete points. In particular, our approach also allows for inequalities, e.g., to enforce the maximum acceleration of a robot at all times. For this reason, we say that our method is more generally applicable to *differential programming*.

Fourth, since our approach reformulates ODEs/PDEs and their boundary conditions as constraints on their control points, they can be combined with other decision variables. This allows us to cast controller design and synthesis problems as optimization problems that don't require expensive simulation. In addition, the nature of the resulting constraints provides insights into the nature of the ODEs/PDEs, allowing us to draw parallels between the mathematical theory of ODEs/PDEs and the computational theory of optimization. In fact, upon reformulation, many interesting classes of ODEs/PDEs yield quadratic programming problems (QPPs) that can be solved in polynomial time.

In this paper, we demonstrate the success of our approach on several interesting classes of ODEs/PDEs. However, given the enormous literature relevant to ODEs/PDEs, our paper can only qualify as a feasibility study in an important direction with preliminary results.

2 Background

In mathematics, Bernstein basis polynomials of degree n are defined to be

$$B_{i,n}(t) = \binom{n}{i} t^i (1-t)^{n-i}, \quad i \in \{0, 1 \dots n\},$$

where $\binom{n}{i}$ is the binomial coefficient equal to $\frac{n!}{i!(n-i)!}$.

A k-dimensional Bézier curve of degree n is of the form

$$\mathbf{B}(t) = \sum_{i=0}^{n} \mathbf{p}_i B_{i,n}(t), \quad t \in [0, 1],$$

where $P = \{\mathbf{p}_0, \mathbf{p}_1 \dots \mathbf{p}_n\}$ is the set of $n+1$ k-dimensional *control points*. Therefore, it is a curve parameterized by t and interpretable as a linear combination of the Bernstein

basis polynomials of degree n. The coefficients of the linear combination are the $n+1$ k-dimensional control points.

A Bernstein polynomial $B(t)$ of degree n is a 1-dimensional Bézier curve of degree n. Therefore, it is a linear combination of the Bernstein basis polynomials of degree n. The coefficients of the linear combination are $n+1$ real numbers acting as 1-dimensional control points.

Bernstein polynomials and Bézier curves have many useful mathematical properties. For example, the Weierstrass Approximation Theorem [11] establishes that any continuous real-valued function defined on the interval $[0,1]$ can be uniformly approximated by Bernstein polynomials.

Two other useful properties of Bernstein polynomials and Bézier curves are with respect to their derivatives and their control points. The derivative of a Bézier curve $\mathbf{B}(t)$ of degree n is a Bézier curve of degree $n-1$. In particular,

$$\frac{d\mathbf{B}(t)}{dt} = \sum_{i=0}^{n-1} \mathbf{p}'_i B_{i,n-1}(t),$$

where the control point $\mathbf{p}'_i = n\,(\mathbf{p}_{i+1} - \mathbf{p}_i)$, for $i \in \{0, 1 \dots n-1\}$.

A Bézier curve $\mathbf{B}(t)$ is bounded by the convex hull of its control points P for $t \in [0,1]$, as shown in Figure 1. Intuitively, this is because for any given value of $t \in [0,1]$: (a) $B_{i,n}(t) \geq 0$ for $i \in \{0, 1 \dots n\}$, and (b) $\sum_{i=0}^{n} B_{i,n}(t) = 1$. Therefore, $\mathbf{B}(t)$ for $t \in [0,1]$ is interpretable as a non-negative linear combination of its control points, necessitating its presence in the convex hull. In particular, $\mathbf{B}(0) = \mathbf{p}_0$ and $\mathbf{B}(1) = \mathbf{p}_n$. In the case of Bernstein polynomials, the control points are 1-dimensional real numbers, and $B(t)$ lies entirely within $[\inf(P), \sup(P)]$.

3 Solving ODEs

To solve ODEs/PDEs using Bézier curves/surfaces, we have to develop the general theory in stride. It is best illustrated through examples and case studies. In this and the next sections, we apply our proposed methodology to a series of examples that are chosen to be in increasing order of complexity and generality. Unless stated otherwise, we use the interval $[0,1]$ for all independent variables since the Bernstein basis polynomials are also defined within the same interval.

3.1 First-Order Homogeneous Linear ODEs with Constant Coefficients

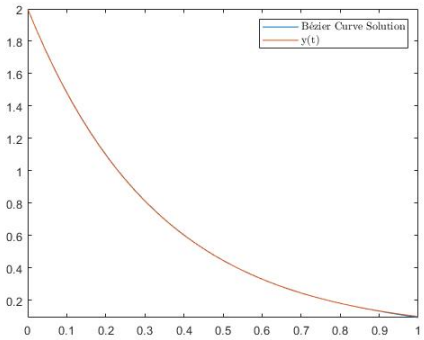
An ODE is said to be *linear* if it is of the form

$$a_0(t)y(t) + a_1(t)y'(t) \dots a_m(t)y^{(m)}(t) + b(t) = 0, \tag{1}$$

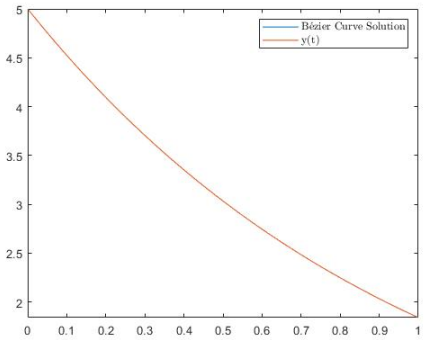
where $a_0(t), a_1(t) \dots a_m(t)$ and $b(t)$ are differentiable functions of t, and the functions $y(t), y'(t) \dots y^{(m)}(t)$ are the successive derivatives of the to-be-determined function $y(t)$.

A first-order linear ODE has $m = 1$. Moreover, a first-order homogeneous linear ODE has $b(t) = 0$. Consider a first-order homogeneous linear ODE with constant coefficients, i.e., $a_0(t)$ and $a_1(t)$ are constants denoted by a_0 and a_1, respectively. Therefore, the ODE is of the form

$$a_0 y(t) + a_1 y'(t) = 0. \tag{2}$$



(a) $3y(t) + y'(t) = 0$; $y(0) = 2$



(b) $y(t) + y'(t) = 0$; $y(0) = 5$

Figure 2 Shows the solutions of two first-order homogeneous linear ODEs with constant coefficients. The solutions generated by the Bézier curve method (blue) match the analytical solutions (orange) exactly. (The blue color is not visible because of the exact match.) The analytical solution for (a) is $y(t) = 2e^{-3t}$. The analytical solution for (b) is $y(t) = 5e^{-t}$. In both cases, 6 control points and 6 test points were used, resulting in a running time of 0.60 s for (a) and 0.45 s for (b).

The above ODE has a known family of solutions of the form $\{Ce^{(-a_0/a_1)t} : C \in \mathbb{R}\}$. A particular solution from this family can be identified based on the initial conditions. Consider the example

$$3y(t) + y'(t) = 0, \tag{3}$$

with the accompanying initial condition $y(0) = 2$.

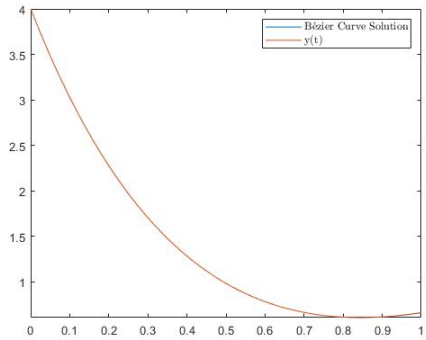
Suppose we represent $y(t)$ using a 1-dimensional Bézier curve $\mathbf{B}(t)$ of degree n and $n+1$ to-be-determined control points $P = \{p_0, p_1 \dots p_n\}$. By substituting $\mathbf{B}(t)$ for $y(t)$ and its derivative $\mathbf{B}'(t)$ for $y'(t)$, the problem reduces to

$$3\sum_{i=0}^{n} p_i B_{i,n}(t) + \sum_{i=0}^{n-1} n(p_{i+1} - p_i) B_{i,n-1}(t) = 0. \tag{4}$$

The initial condition reduces to $p_0 = 2$. In essence, this reduced formulation enforces the polynomial $g(t) = 3\sum_{i=0}^{n} p_i B_{i,n}(t) + \sum_{i=0}^{n-1} n(p_{i+1} - p_i) B_{i,n-1}(t)$ of degree n to be identically equal to 0. Since this can happen only when all the coefficients of the powers of t are individually equal to 0, the problem further reduces to linear equalities on the control points.

Although linear equalities can be solved very efficiently, our first attempt fails for the following reason. We have $n+1$ linear constraints coming from $g(t) \equiv 0$; and we have 1 linear constraint coming from the initial condition. This accounts for a total of $n+2$ linear constraints on $n+1$ variables, creating an over-constrained problem that doesn't necessarily have a solution.

In a second attempt, we split the constraints to *hard* and *soft* constraints. The linear constraints coming from the initial conditions are retained as hard constraints, while the linear constraints coming from $g(t) \equiv 0$ are relaxed to be soft constraints, with a penalty for violation measured using squared error. Of course, the soft constraints should fully incentivize enforcing $g(t) \equiv 0$. Therefore, the squared error is measured on $g(t)$ evaluated at $M \geq n+1$ test points sampled from the interval $[0, 1]$.



(a) $2y(t) + y'(t) = 5t - 3$; $y(0) = 4$

Figure 3 Shows the solution of a first-order non-homogeneous linear ODE with constant coefficients, when $b(t)$ is a polynomial. The solution generated by the Bézier curve method (blue) matches the analytical solution (orange) exactly. (The blue color is not visible because of the exact match.) The analytical solution is $y(t) = \frac{27}{4}e^{-2t} + \frac{5}{2}t - \frac{11}{4}$. Here, 6 control points and 6 test points were used, resulting in a running time of 0.61 s.

Let $t_1, t_2 \dots t_M$ be the test points. We formulate the following QPP:

$$\begin{aligned} \underset{p_0, p_1 \dots p_n}{\text{Minimize}} \quad & \sum_{i=1}^{M} (g(t_i))^2 \\ \text{s.t.} \quad & p_0 = 2. \end{aligned} \tag{5}$$

Solving the QPP yields optimal values of $p_0, p_1 \dots p_n$, which in turn can be used to construct the desired $\mathbf{B}(t)$ as an approximation for $y(t)$. Figure 2(a) shows the Bézier curve solution to the above problem. Figure 2(b) shows the Bézier curve solution to another first-order homogeneous linear ODE with constant coefficients given by

$$y(t) + y'(t) = 0, \tag{6}$$

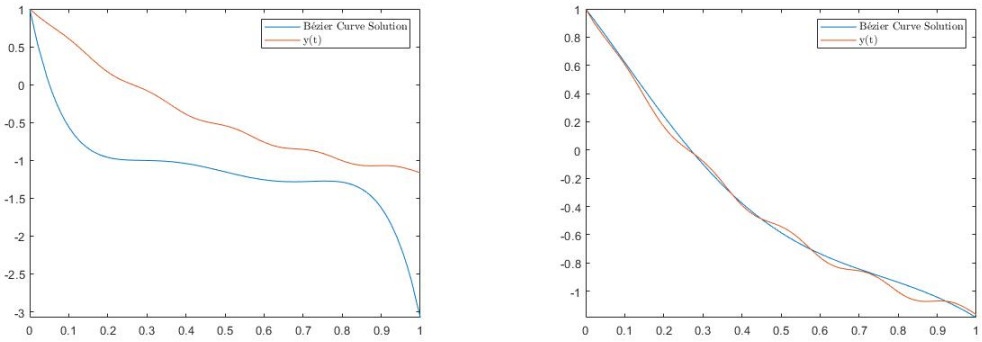
with the accompanying initial condition $y(0) = 5$.

For the QPP solver, we used CVX, a MatLab R2020b package for specifying and solving convex programs. We report the CVX running times for all examples discussed in this paper. All experiments were conducted on a laptop with a 2.8GHz Quad-Core Intel Core i7 processor and 16GB 2133MHz DDR4 memory. We used the default CVX settings for all experiments.

3.2 First-Order Non-Homogeneous Linear ODEs with Constant Coefficients

We now examine first-order non-homogeneous linear ODEs with constant coefficients. This is similar to the previous subsection, except that $b(t)$ is not necessarily 0. We refer to a non-zero $b(t)$ as the non-homogeneity term.

If $b(t)$ is a polynomial, the formulation using Bézier curves again reduces to a case of polynomial equivalence, and our proposed methodology continues to be directly applicable. However, we note that if the degree of $b(t)$ exceeds n, then the number of test points M should be $\geq \deg(b(t)) + 1$.



(a) $2y(t) + y'(t) = \sin(30t) - 3;\ y(0) = 1$ **(b)** $2y(t) + y'(t) = \sin(30t) - 3;\ y(0) = 1$

Figure 4 Shows the solutions of a first-order non-homogeneous linear ODE with constant coefficients, when $b(t)$ is not a polynomial. The solution generated by the Bézier curve method (blue) better approximates the analytical solution (orange) with an increasing number of test points. The analytical solution is $y(t) = (1145e^{-2t} + \sin(30t) - 15\cos(30t) - 678)/452$. In (a), 6 control points and 6 test points were used, resulting in a running time of 0.38 s. In (b), 6 control points and 20 test points were used, resulting in a running time of 0.57 s.

Consider the following example:

$$2y(t) + y'(t) = 5t - 3 \tag{7}$$
$$y(0) = 4.$$

The analytical solution of this ODE is given by $y(t) = \frac{27}{4}e^{-2t} + \frac{5}{2}t - \frac{11}{4}$. Figure 3 shows this analytical solution and the Bézier curve solution obtained using $n = 5$ and $M = 6$. Once again, the Bézier curve solution provides very accurate results.

If $b(t)$ is not a polynomial, the formulation is not reducible to one of polynomial equivalence. Nonetheless, our method can still be used since Bézier curves can uniformly approximate any function [11].

Consider the following example:

$$2y(t) + y'(t) = \sin(30t) - 3 \tag{8}$$
$$y(0) = 1.$$

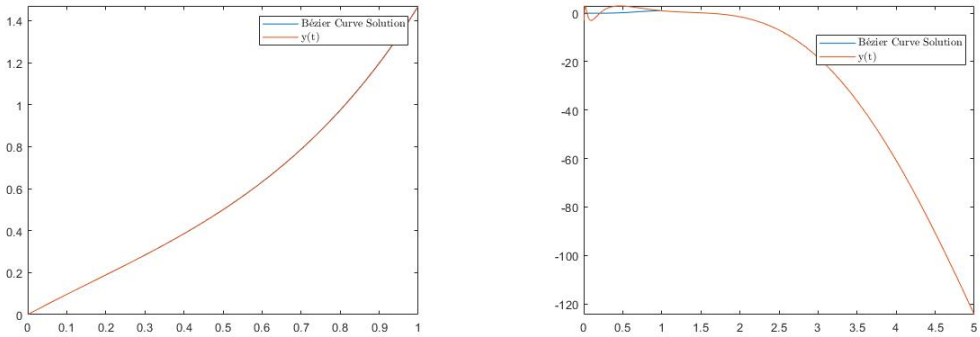
The non-homogeneity term $b(t)$ is no longer a polynomial and is in fact very oscillatory. The analytical solution of this ODE is given by $y(t) = (1145e^{-2t} + \sin(30t) - 15\cos(30t) - 678)/452$. Figure 4(a) shows that the Bézier curve solution has a large deviation from the analytical solution when $n = 5$ and $M = 6$. However, Figure 4(b) shows that the accuracy of the Bézier curve solution improves significantly as M increases, i.e., when $n = 5$ and $M = 20$.

3.3 Higher-Order Linear ODEs

In this subsection, we discuss higher-order linear ODEs. These are linear ODEs with $m > 1$. They can be classified as homogeneous or non-homogeneous, depending on $b(t)$. If $b(t) = 0$, the ODE is homogeneous; otherwise, it is non-homogeneous with $b(t)$ referred to as the non-homogeneity term. We consider two illustrative types of higher-order linear ODEs.

Consider an ODE of the form

$$a_0 y(t) + a_1 y'(t) + a_2 y''(t) = 0, \tag{9}$$



(a) $-6y(t) + y'(t) + y''(t) = 0;\ y(0) = 0, y'(0) = 1$ **(b)** $\frac{13}{t^2}y(t) - \frac{5}{t}y'(t) + y''(t) = 0;\ y(1) = 1, y'(1) = 0$

Figure 5 Shows the solutions of two higher-order linear ODEs. (a) shows an ODE with constant coefficients, and (b) shows an Euler-Cauchy ODE. In (a), the solution generated by the Bézier curve method (blue) matches the analytical solution (orange) exactly. (The blue color is not visible because of the exact match.) In (b), the solution generated by the Bézier curve method (blue) approximates the analytical solution (orange). The analytical solution for (a) is $y(t) = \frac{1}{5}(e^{2t} - e^{-3t})$. The analytical solution for (b) is $y(t) = 0.5t^3(2\cos(2\log t) - 3\sin(2\log t))$. In (a), 6 control points and 6 test points were used, resulting in a running time of 0.45 s. In (b), 6 control points and 20 test points were used, resulting in a running time of 0.71 s.

where a_0, a_1 and a_2 are constants. This type of ODE can be solved using the method of characteristic equations, i.e., by finding the roots of the polynomial $a_0 + a_1\lambda + a_2\lambda^2 = 0$ and using them to form linearly independent solutions of the ODE.

We can apply our Bézier curve method to this class of ODEs as well. This is because the second derivative $\frac{d^2\mathbf{B}(t)}{dt^2}$ of a Bézier curve $\mathbf{B}(t)$ is also a Bézier curve. If $\mathbf{B}(t)$ is of degree n, $\frac{d^2\mathbf{B}(t)}{dt^2}$ is of degree $n-2$. Moreover, the control points of $\frac{d^2\mathbf{B}(t)}{dt^2}$ are simple linear combinations of the control points of $\mathbf{B}(t)$.

When $a_0 = -6$, $a_1 = 1$ and $a_2 = 1$, the analytical solution is given by $y(t) = \frac{1}{5}(e^{2t} - e^{-3t})$ for the initial conditions $y(0) = 0, y'(0) = 1$. Figure 5(a) shows the solution generated by the Bézier curve method. This solution matches the analytical solution exactly.

Now consider the Euler-Cauchy ODE of the form

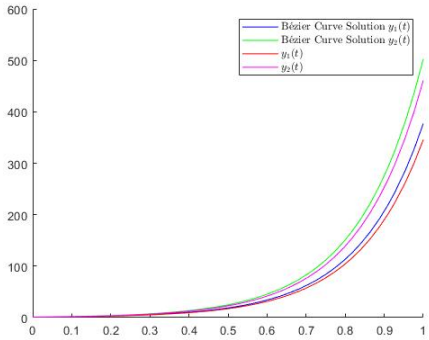
$$\frac{q}{t^2}y(t) + \frac{p}{t}y'(t) + y''(t) = 0, \tag{10}$$

where p and q are constants. This class of ODEs also has well-studied analytical methods for finding general solutions. When $p = -5$ and $q = 13$, the analytical solution is given by $y(t) = 0.5t^3(2\cos(2\log t) - 3\sin(2\log t))$ for the initial conditions $y(1) = 1, y'(1) = 0$. Figure 5(b) shows the solution generated by the Bézier curve method. This solution approximates the analytical solution fairly well.

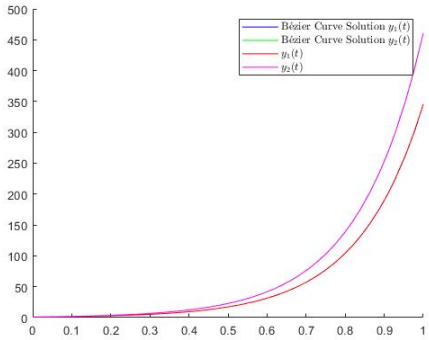
The Bézier curve solution of the Euler-Cauchy ODE improves with increasing n and M. Similarly, the Bézier curve approximations improve with increasing n and M when $b(t)$ is oscillatory.

4 Solving Systems of ODEs

In this section, we apply our methods to systems of ODEs. In such cases, we have to solve for a vector of unknown functions $\bar{y}(t)$ that satisfy differential equations involving their



(a) $\bar{y}'(t) = A\bar{y}(t)$; $\bar{y}(0) = (1,1)^\top$



(b) $\bar{y}'(t) = A\bar{y}(t)$; $\bar{y}(0) = (1,1)^\top$

Figure 6 Shows the solutions of a system of ODEs. The solution generated by the Bézier curve method (blue and green for $y_1(t)$ and $y_2(t)$, respectively) better approximates the analytical solution (red and pink, respectively) with increasing n. The analytical solution is $\bar{y}(t) = \frac{1}{7}(1,-1)^\top e^{-t} + \frac{2}{7}(3,4)^\top e^{6t}$. In (a), 8 control points and 20 test points were used, resulting in a running time of 1.03 s. In (b), 11 control points and 20 test points were used, resulting in a running time of 0.90 s. (The blue and green colors are not visible because of the exact match.) The columns of A are $(2,4)^\top$ and $(3,3)^\top$ in that order.

derivatives. Although systems of ODEs invoke matrices to describe the relationships between the various unknown functions and their derivatives, they involve only one independent variable t. We focus our discussion on an example that illustrates the generality of our Bézier curve method.

Consider the system of ODEs

$$\bar{y}'(t) = A\bar{y}(t), \tag{11}$$

where $\bar{y}(t) : \mathbb{R} \to \mathbb{R}^2$ and A is a 2×2 matrix of real numbers. We have to solve for $\bar{y}(t) = (y_1(t), y_2(t))^\top$.

The family of solutions for this system of ODEs is intimately related to the eigenvalues of A. If A has two distinct real eigenvalues, λ_1 and λ_2, with corresponding eigenvectors $\bar{v}_1$ and $\bar{v}_2$, the general solution is given by $C_1\bar{v}_1e^{\lambda_1 t} + C_2\bar{v}_2e^{\lambda_2 t}$, for constants C_1 and C_2. If A has complex conjugate eigenvalues, $\lambda_1 \pm \mathbf{i}\lambda_2$, with corresponding eigenvectors $\bar{v}_1 \pm \mathbf{i}\bar{v}_2$, the general solution is given by $C_1(\bar{v}_1 \cos(\lambda_2 t) - \bar{v}_2 \sin(\lambda_2 t))e^{\lambda_1 t} + C_2(\bar{v}_1 \sin(\lambda_2 t) + \bar{v}_2 \cos(\lambda_2 t))e^{\lambda_1 t}$. If A has a repeated real eigenvalue λ and the eigenvectors $\bar{v}_1$ and $\bar{v}_2$ are linearly independent, the general solution is given by $C_1\bar{v}_1e^{\lambda t} + C_2\bar{v}_2e^{\lambda t}$. If A has only one linearly independent eigenvector $\bar{v}$, the general solution is given by $C_1\bar{v}e^{\lambda t} + C_2(\bar{v}te^{\lambda t} + \bar{\eta}3^{\lambda t})$, where $\bar{\eta}$ is any solution of $(A - \lambda I)\bar{\eta} = \bar{v}$.

For illustration, suppose $A = \begin{pmatrix} 2 & 3 \\ 4 & 3 \end{pmatrix}$. Its eigenvalues are $\lambda_1 = -1$ and $\lambda_2 = 6$, with corresponding eigenvectors $\bar{v}_1 = (1,-1)^\top$ and $\bar{v}_2 = (3,4)^\top$. When accompanied by the initial condition $\bar{y}(0) = (1,1)^\top$, the analytical solution is $\bar{y}(t) = \frac{1}{7}(1,-1)^\top e^{-t} + \frac{2}{7}(3,4)^\top e^{6t}$.

We can also solve for $\bar{y}(t)$ using our Bézier curve method. The idea is to represent it as a Bézier curve $\mathbf{B}(t)$ with 2-dimensional control points. If $\mathbf{B}(t)$ is chosen to be of degree n, it has $n+1$ to-be-determined control points $P = \{\bar{p}_0, \bar{p}_1 \ldots \bar{p}_n\}$. Using the test points

$t_1, t_2 \dots t_M$, we formulate the following QPP:

$$\underset{\bar{p}_0, \bar{p}_1 \dots \bar{p}_n}{\text{Minimize}} \quad \sum_{i=1}^{M} \epsilon_i^2 \tag{12}$$

$$\text{s.t.} \quad \mathbf{B}(0) = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

$$\forall 1 \le i \le M: \quad \begin{pmatrix} -\epsilon_i \\ -\epsilon_i \end{pmatrix} \le \mathbf{B}'(t_i) - A\mathbf{B}(t_i) \le \begin{pmatrix} \epsilon_i \\ \epsilon_i \end{pmatrix}.$$

The constraints in this problem are linear since $\mathbf{B}(t)$ and $\mathbf{B}'(t)$ yield linear combinations of the to-be-determined control points when evaluated at a specific t.

Figure 6 shows the solutions generated by the Bézier curve method. The solutions approximate the analytical solution very well; and the accuracy increases with increasing n and M.

5 Solving PDEs

In this section, we apply our methods to PDEs. In such cases, we have multiple independent variables; and the required function is a surface in high-dimensional space. The differential equations specifying the characteristics of the required function can involve its partial derivatives. We generalize our Bézier *curve* method to the Bézier *surface* method. For illustration, we focus our discussion on solving PDEs for a function $f(t, u)$ on two independent variables t and u.

A k-dimensional Bézier surface $\mathbf{B}(t, u)$ of degrees $n_t \times n_u$ is characterized by the k-dimensional control points $\mathbf{p}_{i,j}$, for $0 \le i \le n_t$ and $0 \le j \le n_u$. It is given by

$$\mathbf{B}(t, u) = \sum_{i=0}^{n_t} \sum_{j=0}^{n_u} \mathbf{p}_{i,j} B_{i,n_t}(t) B_{j,n_u}(u), \tag{13}$$

where $B_{i,n_t}(t)$ and $B_{j,n_u}(u)$ are the Bernstein basis polynomials. The partial derivatives of $\mathbf{B}(t, u)$ are given by

$$\frac{\partial \mathbf{B}(t,u)}{\partial t} = n_t \sum_{i=0}^{n_t-1} \sum_{j=0}^{n_u} (\mathbf{p}_{i+1,j} - \mathbf{p}_{i,j}) B_{i,n_t-1}(t) B_{j,n_u}(u)$$

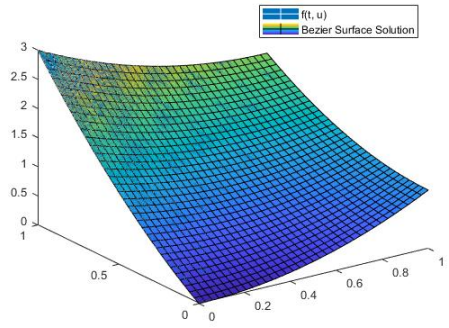
$$\frac{\partial \mathbf{B}(t,u)}{\partial u} = n_u \sum_{j=0}^{n_u-1} \sum_{i=0}^{n_t} (\mathbf{p}_{i,j+1} - \mathbf{p}_{i,j}) B_{i,n_t}(t) B_{j,n_u-1}(u)$$

Bézier surfaces have attractive mathematical properties equivalent to those of Bézier curves [17]. These include their ability to approximate any surface with a sufficient number of control points, being closed under the operations of differentiation, and being entirely within the convex hull of their control points. For a function $f(t, u)$ represented as a Bézier surface $\mathbf{B}(t, u)$ of degrees $n_t \times n_u$, there are $(n_t + 1) \times (n_u + 1)$ to-be-determined control points. Evaluating $\mathbf{B}(t, u)$ at a specific test point (t, u) yields a linear combination of these control points that can be easily incorporated into the formulation of a QPP.

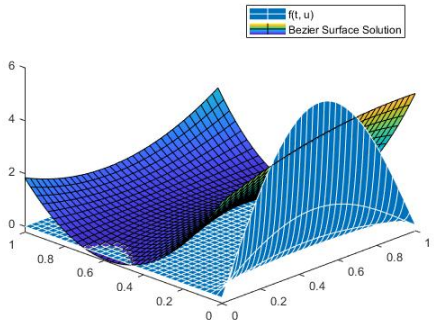
Consider the following example PDE:

$$f_t(t, u) + f_u(t, u) + 2 = 0 \tag{14}$$

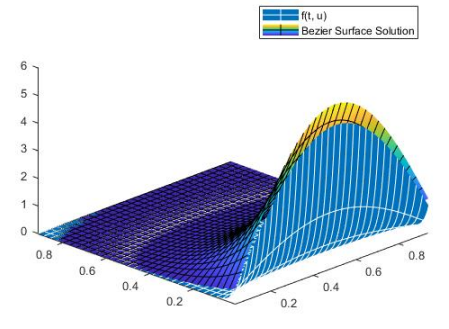
$$f(t, 0) = t^2,$$



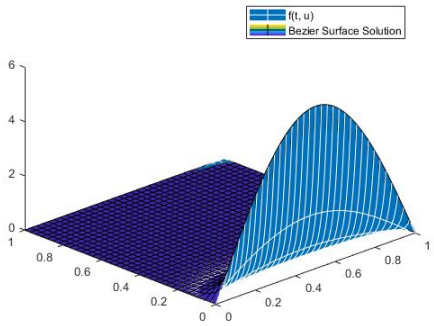
(a) $f_t(t,u) + f_u(t,u) + 2 = 0;\ f(t,0) = t^2$



(b) $f_t(t,u) - 4f_{uu}(t,u) = 0;\ f(0,u) = 6\sin(\pi u), f(t,0) = 0, f(t,1) = 0$



(c) $f_t(t,u) - 4f_{uu}(t,u) = 0;\ f(0,u) = 6\sin(\pi u), f(t,0) = 0, f(t,1) = 0$



(d) $f_t(t,u) - 4f_{uu}(t,u) = 0;\ f(0,u) = 6\sin(\pi u), f(t,0) = 0, f(t,1) = 0$

Figure 7 Shows the solutions of some PDEs. The solutions generated by the Bézier surface method are good approximations to the analytical solutions. The quality of the solutions increases with the number of test points and the degrees used in the Bézier surface. In (a), the analytical solution is $f(t,u) = 2u + (t-u)^2$. In (b)-(d), the analytical solution is $f(t,u) = 6\sin(\pi u)e^{-4\pi^2 t}$. In (a), 4×4 control points and 17 test points were used, resulting in a running time of 2.36 s. In (b), 3×3 control points and 10 test points were used, resulting in a running time of 1.85 s. In (c), 6×6 control points and 37 test points were used, resulting in a running time of 2.33 s. In (d), 11×11 control points and 122 test points were used, resulting in a running time of 20.45 s. In (a) and (d), the Bézier surface solution is an exact match to the analytical solution.

where $f_t(t,u)$ and $f_u(t,u)$ denote the partial derivatives of $f(t,u)$ with respect to t and u, respectively, and $f(t,0) = t^2$ is a boundary condition.

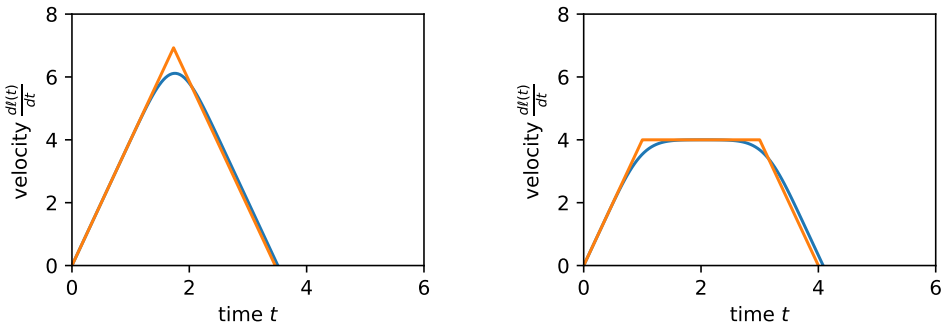
Using the test points $(t_1,u_1),(t_2,u_2)\ldots(t_M,u_M)$, we formulate the following QPP:

$$\underset{p_{i,j}:0\le i\le n_t, 0\le j\le n_u}{\text{Minimize}} \quad \sum_{l=1}^{M}(\epsilon_l^2 + \varepsilon_l^2) \quad \text{s.t.} \tag{15}$$

$$\forall 1 \le l \le M: \quad -\epsilon_l \le \frac{\partial \mathbf{B}(t,u)}{\partial t} + \frac{\partial \mathbf{B}(t,u)}{\partial u} + 2 \le \epsilon_l$$

$$\forall 1 \le l \le M: \quad -\varepsilon_l \le \mathbf{B}(t_l,0) - t_l^2 \le \varepsilon_l.$$

Our test points are chosen from $[0,1]\times[0,1]$. Unlike the initial conditions in ODEs that were imposed as hard constraints at specific points, the boundary conditions in PDEs typically



(a) $v_{\max} = 8\,\mathrm{m/s}$ **(b)** $v_{\max} = 4\,\mathrm{m/s}$

Figure 8 Shows the result of applying our Bézier curve method to the motion profile problem. The orange curves are the optimal velocity profiles derived from physical interpretation. The blue curves are close approximations produced by our method. In both cases, 40 control points were used for the approximation. In both (a) and (b), $L = 12\,\mathrm{m}$, $a_{\max} = 4\,\mathrm{m/s^2}$ and $a_{\min} = -4\,\mathrm{m/s^2}$. In (a), $v_{\max} = 8\,\mathrm{m/s}$, and in (b), $v_{\max} = 4\,\mathrm{m/s}$.

involve entire subspaces. For example, the boundary condition $f(t,0) = t^2$ involves all $t \in [0,1]$. While we can express this condition as a hard constraint equating two polynomials, it risks posing an over-constrained problem. Therefore, we include it as a soft constraint in the objective function, with each test point contributing a term to it.

Figure 7(a) shows the result of our method for the above PDE with $n_t = 3$, $n_u = 3$ and $M = 17$. The result is an exact match to the known analytical solution $f(t,u) = 2u + (t-u)^2$.

We now consider a popular Heat Equation widely used in physics [18]. With Dirichlet boundary conditions, the PDE is as follows:

$$\begin{aligned} &f_t(t,u) - 4f_{uu}(t,u) = 0 \\ &f(0,u) = 6\sin(\pi u) \\ &f(t,0) = 0, f(t,1) = 0. \end{aligned} \tag{16}$$

Here, $f_{uu}(t,u)$ refers to the second partial derivative $\frac{\partial^2 f(t,u)}{\partial u^2}$. The analytical solution is given by $f(t,u) = 6\sin(\pi u)e^{-4\pi^2 t}$.

Figures 7(b)-(d) show the results of our method for the Heat Equation with different values of n_t, n_u and M. The accuracy improves with increasing degrees and number of test points. In fact, an exact match to the analytical solution is achieved relatively quickly, as shown in Figure 7(d).

6 Differential Programming with Inequalities

In the foregoing sections, we demonstrated the viability of our approach on various kinds of ODEs and PDEs. As already outlined in the Introduction, we conducted this feasibility study in anticipation of reaping the many benefits of our approach compared to existing methods. In this section, we show one such benefit in allowing the use of inequalities.

Inequalities and differential operators are commonplace in robotics, physics, and hybrid systems, among many other areas of science and engineering. For example, in robotics, a robot might have a maximum acceleration or deceleration capability that is posed as an inequality involving the second derivative of its motion profile. Existing analytical techniques

are not capable of handling inequalities; and existing numerical techniques do not produce an analytical solution that may be required for downstream tasks. However, our Bézier curve method is viable in such situations.

Consider the following simple motivating example. Suppose a robot is required to travel a distance L in a straight line between two points A and B. Suppose it is required to start at A and end at B with 0 velocities; and suppose it has maximum velocity $v_{\max} \geq 0$, maximum acceleration $a_{\max} \geq 0$ and minimum acceleration $a_{\min} \leq 0$. The goal is to minimize the traversal time T. Intuitively, the optimal solution is to start with maximum acceleration $a_{\max}$ and stop with maximum deceleration $|a_{\min}|$. In between, the robot should cap off at the maximum velocity $v_{\max}$. The two possible scenarios are illustrated in Figure 8.

Stated purely mathematically, the differential programming problem involves inequalities and is as follows:

$$\begin{aligned} &\text{Find} \quad \ell(t) : [0, T] \to \mathbb{R} \text{ and minimum } T \quad \text{s.t.} \\ &\forall t : \quad \ell'(t) \leq v_{\max} \\ &\forall t : \quad a_{\min} \leq \ell''(t) \leq a_{\max} \\ &\qquad \ell(0) = 0, \ell(T) = L \\ &\qquad \ell'(0) = 0, \ell'(T) = 0. \end{aligned} \tag{17}$$

As such, solving this mathematical problem without the physical interpretation is not straightforward even from the perspective of techniques available in calculus. This is primarily because of the inequalities imposed on the derivatives of continuous functions.

In contrast, our Bézier curve method solves this problem efficiently since inequalities are naturally allowed in OR solvers. Figure 8 shows the Bézier curve solutions for $\ell(t)$, the distance function that represents the distance covered at time t starting from A, for the two possible scenarios. (See [19] for more details on this approach to solve the motion profile problem and its generalization to the multi-robot scenario.)

We also note that our Bézier curve method is not just any polynomial-fitting method. General polynomial-fitting methods cannot enforce global conditions on a function since they are required to hold for *all* t. In our method, the convex hull property of Bézier curves is invoked to ensure that satisfying inequalities at only the control points entails that they are also globally satisfied.

7 Discussion

There are many anticipated benefits of our approach since it casts differential operators in the language of OR, and consequently, in the language of search. Many optimization problems in science and engineering that may not be directly amenable to analytical methods can instead be solved programmatically using OR solvers. In turn, powerful OR solvers like Gurobi are scalable to millions of variables. They also employ efficient parallelization techniques. Moreover, since OR is already being studied in relation to constraint programming (CP) and artificial intelligence (AI), our framework paves the way for combining the strengths of variational techniques used in calculus, primal-dual techniques used in OR, constraint propagation techniques used in CP, and heuristic search techniques used in AI.

Many problems in the real world can also benefit from rendering differential operators in the language of search. In addition to scalability and reasoning with inequalities, this reformulation allows us to introduce extra decision variables. Testing and verification of complex systems involving ODEs/PDEs can be done via highly scalable search-based

methods instead of prohibitively expensive simulation-based methods. Optimization problems in computational physics, e.g., how to optimally place a set of teflon dielectric cylinders to focus electromagnetic power, can be solved using search-based methods after rendering the PDEs of electromagnetism in the language of OR.

8 Conclusions and Future Work

In this paper, we presented an OR-based approach for differential programming to address the drawbacks of existing analytical and numerical methods. Analytical methods mostly remain an art and are largely insufficient for complex systems. Numerical approximation methods do not yield a viable analytical form of the solution that is often required for downstream tasks. Our main idea was to represent entire functions as Bézier curves or Bézier surfaces with to-be-determined control points. The ODEs/PDEs as well as their boundary conditions can then be reformulated as constraints on these control points. In many cases, we showed that this reformulation yields QPPs that can be solved efficiently. We also demonstrated the use of our approach in differential programming with inequalities.

More generally, our work facilitates search-based methods for solving problems that involve differential operators and sets the stage for combining the strengths of variational methods used in calculus and search-based pruning methods used in OR, CP and AI. There are many avenues of future work based on the foregoing discussions. We are also interested in the idea of representing local regions of functions using separate Bézier curves/surfaces and "stitching" them together under conditions of continuity to achieve more efficiency.

References

1. Ji-wung Choi, Renwick Curry, and Gabriel Elkaim. Path planning based on bézier curve for autonomous ground vehicles. In *Advances in Electrical and Electronics Engineering - IAENG Special Edition of the World Congress on Engineering and Computer Science*, pages 158–166, 2008.
2. Ken Dill and Sarina Bromberg. *Molecular Driving Forces: Statistical Thermodynamics in Biology, Chemistry, Physics, and Nanoscience*. Garland Science, 2012.
3. Robert Eymard, Thierry Gallouët, and Raphaèle Herbin. Finite volume methods. *Handbook of Numerical Analysis*, 7:713–1018, 2000.
4. N. Fallah and N. Nikraftar. Meshless finite volume method for the analysis of fracture problems in orthotropic media. *Engineering Fracture Mechanics*, 204:46–62, 2018.
5. Gerald Farin. *Curves and Surfaces for Computer-Aided Geometric Design: A Practical Guide*. Elsevier, 2014.
6. Torkel Glad and Lennart Ljung. *Control Theory*. CRC Press, 2018.
7. Antonio Huerta, Ted Belytschko, Sonia Fernández-Méndez, Timon Rabczuk, Xiaoying Zhuang, and Marino Arroyo. Meshfree methods. *Encyclopedia of Computational Mechanics (Second Edition)*, pages 1–38, 2018.
8. Thomas Hughes, Giancarlo Sangalli, and Mattia Tani. *Isogeometric Analysis: Mathematical and Implementational Aspects, with Applications*. Lecture Notes in Mathematics Book Series (LNM, volume 2219), 2018.
9. I. Krasilshchik and A. Vinogradov. Nonlocal trends in the geometry of differential equations: Symmetries, conservation laws, and bäcklund transformations. In *Symmetries of Partial Differential Equations*, pages 161–209. Springer, 1989.
10. Tadeusz Liszka and Janusz Orkisz. The finite difference method at arbitrary irregular grids and its application in applied mechanics. *Computers & Structures*, 11(1-2):83–95, 1980.
11. George Lorentz. *Bernstein Polynomials*. American Mathematical Soc., 1986.

446 **12** Michael Mortenson. *Mathematics for Computer Graphics Applications*. Industrial Press Inc.,
447 1999.
448 **13** Lev Ovsiannikov. *Group Analysis of Differential Equations*. Academic Press, 2014.
449 **14** Michael Renardy and Robert Rogers. *An Introduction to Partial Differential Equations*.
450 Springer, 2006.
451 **15** Alberto Rojo and Anthony Bloch. *The Principle of Least Action: History and Physics*.
452 Cambridge University Press, 2018.
453 **16** Sarah Tang and Vijay Kumar. Safe and complete trajectory generation for robot teams with
454 higher-order dynamics. In *IEEE/RSJ International Conference on Intelligent Robots and*
455 *Systems*, pages 1894–1901, 2016.
456 **17** Lianqiang Yang and Xiao-Ming Zeng. Bézier curves and surfaces with shape parameters.
457 *International Journal of Computer Mathematics*, 86:1253–1263, 2009.
458 **18** Hugh Young, Roger Freedman, and Albert Ford. *Sears and Zemansky's University Physics*.
459 Pearson Addison-Wesley, 2006.
460 **19** Han Zhang, Neelesh Tiruviluamala, Sven Koenig, and T. K. Satish Kumar. Temporal reasoning
461 with kinodynamic networks. In *Proceedings of the Thirty-First International Conference on*
462 *Automated Planning and Scheduling*, 2021.
463 **20** Olgierd Zienkiewicz, Robert Taylor, Perumal Nithiarasu, and J. Zhu. *The Finite Element*
464 *Method*. McGraw-Hill London, 1977.