Mathematical Optimization – Toán Tối Ưu

Nguyễn Quản Bá Hồng*

Ngày 17 tháng 10 năm 2024

Tóm tắt nội dung

This text is a part of the series *Some Topics in Advanced STEM & Beyond*:
URL: https://nqbh.github.io/advanced_STEM/.
Latest version:

- *Mathematical Optimization – Toán Tối Ưu.*
  PDF: URL: https://github.com/NQBH/advanced_STEM_beyond/blob/main/optimization/NQBH_mathematical_optimization.pdf.
  TEX: URL: https://github.com/NQBH/advanced_STEM_beyond/blob/main/optimization/NQBH_mathematical_optimization.tex.

Mục lục

1 Basic

2 Linear Programming – Quy Hoạch Tuyến Tính

Definition 1 (Linear programming)**.** "Linear programming (LP), *also called* linear optimization, *is a method to achieve the best outcome, e.g., maximum profit or lower cost, in a mathematical model whose requirements & objective are represented by linear relationships. Linear programming is a special case of mathematical programming* ≡ *mathematical optimization.*" – *Wikipedia/linear programming*

More formally, linear programming is a technique for the optimization of a linear linear objective function, subject to linear equality & linear inequality constraints. Its feasible region is a convex polytope, which is a set defined as the intersection of finitely many half spaces, each of which is defined by a linear inequality. Its objective function is a real-valued affine (linear) function defined on this polytope. A linear programming algorithm finds a point in the polytope where this function has the largest (or smallest) value if such a point exists.

*A Scientist & Creative Artist Wannabe. E-mail: nguyenquanbahong@gmail.com. Bến Tre City, Việt Nam.

Linear programs are problems that can be expressed in standard form as

$$\text{Find a vector } \mathbf{x} \text{ that maximizes/minimizes } \mathbf{c}^\top \mathbf{x} \text{ subject to } A\mathbf{x} \leq \mathbf{b} \ \& \ \mathbf{x} \geq \mathbf{0}. \tag{lp}$$

Here the components of $\mathbf{x}$ are the variables to be determined, $\mathbf{b}, \mathbf{c}$ are given vectors, & A is a given matrix. The function whose value is to be maximized ($\mathbf{x} \mapsto \mathbf{c}^\top \mathbf{x}$ in this case) is called the objective function. The constraint $A\mathbf{x} \leq \mathbf{x}$ & $\mathbf{x} \geq \mathbf{0}$ specify a convex polytope over which the objective function is to be optimized.

Linear programming can be applied to various fields of study, which is widely used in mathematics &, to a lesser extent, in business, economics, & to some engineering problems. There is a close connection between linear programs, eigenequations, John von Neumann's general equilibrium model, & structural equilibrium models (see dual linear program). Industries using linear programming models include transportation, energy, telecommunications, & manufacturing. It has proven useful in modeling diverse types of problems in planning, routing, scheduling, assignment, & design.

Định nghĩa 1 (Quy hoạch tuyến tính)**.** *Bài toán* quy hoạch tuyến tính *là bài toán tìm* GTLN/GTNN *của* hàm mục tiêu *trong điều kiện hàm mục tiêu là hàm bậc nhất đối với các biến & mỗi 1 điều kiện ràng buộc là bất phương trình bậc nhất đối với các biến (không kể điều kiện ràng buộc biến thuộc tập số nào, e.g.,* $\mathbb{N}, \mathbb{Q}, \mathbb{R}, \mathbb{C}$.

Ta có thể viết bài toán quy hoạch tuyến tính 2 biến x, y về dạng sau:

$$\max T := \alpha x + \beta y \text{ s.t } a_i x + b_i y \leq c_i, \ \forall i = 1, 2, \ldots, n, \tag{lp2max}$$

$$\min T := \alpha x + \beta y \text{ s.t } a_i x + b_i y \leq c_i, \ \forall i = 1, 2, \ldots, n, \tag{lp2min}$$

trong đó các điều kiện ràng buộc đều là các bất phương trình bậc nhất đối với x, y. See also:

- *Problem: Inequation & Linear System of Inequations – Bài Tập: Bất Phương Trình & Hệ Bất Phương Trình.*
  Folder: Elementary STEM & Beyond/Elementary Mathematics/grade 10/linear system inequations/problem: [pdf[1]][TEX[2]].
  - *Problem & Solution: Inequation & Linear System of Inequations – Bài Tập & Lời Giải: Bất Phương Trình & Hệ Bất Phương Trình.*
    Folder: Elementary STEM & Beyond/Elementary Mathematics/grade 10/linear system inequations/solution: [pdf[3]][TEX[4]].
- *Problem: Mathematical Optimization – Bài Tập: Ứng Dụng Toán Học Để Giải Quyết 1 Số Bài Toán Tối Ưu.*
  Folder: Elementary STEM & Beyond/Elementary Mathematics/grade 12/optimization/problem: [pdf[5]][TEX[6]].
  - *Problem & Solution: Mathematical Optimization – Bài Tập & Lời Giải: Ứng Dụng Toán Học Để Giải Quyết 1 Số Bài Toán Tối Ưu.*
    Folder: Elementary STEM & Beyond/Elementary Mathematics/grade 12/optimization/solution: [pdf[7]][TEX[8]].

2.1 How to solve some linear programmings – Cách giải 1 số bài toán quy hoạch tuyến tính

Có thể giải 1 số bài toán quy hoạch tuyến tính dạng (lp2max) hay (lp2min) theo 2 bước:

1. Xác định miền nghiệm $S \subset \mathbb{R}^2$ của hệ bất phương trình $a_i x + b_i y \leq c_i$, $\forall i = 1, \ldots, n$.
2. Tìm điểm $(x, y) \in S$ sao cho biểu thức $T = T(x, y) = \alpha x + \beta y$ có GTLN ở bài toán (lp2max) hoặc có GTNN ở bài toán (lp2min).

   Khi miền nghiệm S là đa giác (polygon), biểu thức $T(x, y) = \alpha x + \beta y$ đạt GTLN/GTNN (gộp chung gọi là *cực trị*) tại $(x, y) \in \mathbb{R}^2$ là tọa độ 1 trong các đỉnh của đa giác đó. Khi đó, bước 2 có thể được thực hiện như sau:

   (a) Xác định tọa độ các đỉnh của đa giác đó.

   (b) Tính giá trị của biểu thức $T(x, y) = \alpha x + \beta y$ tại các đỉnh của đa giác đó.

   (c) So sánh các giá trị & kết luận.

[Thá+25, Chuyên đề II, §1, LT1–3, 1., 2., 3., 4., 5., pp. 20–25].

[1] URL: https://github.com/NQBH/elementary_STEM_beyond/blob/main/elementary_mathematics/grade_10/linear_system_inequations/problem/NQBH_linear_system_inequations_problem.pdf.
[2] URL: https://github.com/NQBH/elementary_STEM_beyond/blob/main/elementary_mathematics/grade_10/linear_system_inequations/problem/NQBH_linear_system_inequations_problem.tex.
[3] URL: https://github.com/NQBH/elementary_STEM_beyond/blob/main/elementary_mathematics/grade_10/linear_system_inequations/solution/NQBH_linear_system_inequations_solution.pdf.
[4] URL: https://github.com/NQBH/elementary_STEM_beyond/blob/main/elementary_mathematics/grade_10/linear_system_inequations/solution/NQBH_linear_system_inequations_solution.tex.
[5] URL: https://github.com/NQBH/elementary_STEM_beyond/blob/main/elementary_mathematics/grade_12/optimization/problem/NQBH_optimization_problem.pdf.
[6] URL: https://github.com/NQBH/elementary_STEM_beyond/blob/main/elementary_mathematics/grade_12/optimization/problem/NQBH_optimization_problem.tex.
[7] URL: https://github.com/NQBH/elementary_STEM_beyond/blob/main/elementary_mathematics/grade_12/optimization/solution/NQBH_optimization_solution.pdf.
[8] URL: https://github.com/NQBH/elementary_STEM_beyond/blob/main/elementary_mathematics/grade_12/optimization/solution/NQBH_optimization_solution.tex.

3 Optimal Control – Điều Khiển Tối Ưu

Resources – Tài nguyên.

1. [Lio69]. Jacques-Louis Lions. *Quelques méthodes de résolution des problèmes aux limites nonlinéaires* – Some Methods for Solving Nonlinear Boundary Problems.

2. [Lio71]. Jacques-Louis Lions. *Optimal control of systems governed by PDEs.*

3. [Trö10]. Fredi Tröltzsch. *Optimal Control of PDEs.*

Definition 2 (Optimal control theory)**.** "Optimal control theory *is a branch of control theory that deals with finding a control for a dynamical system over a period of time s.t. an objective function is optimized. It has numerous applications in science, engineering, & operational research. E.g., the dynamical system might be a spacecraft with controls corresponding to rocket thrusters, & the objective might be to reach the Moon with minimum fuel expenditure. Or the dynamical system could be a nation's economy, with the objective to minimize unemployment; the controls in this case could be fiscal & monetary policy. A dynamical system may also be introduced to embed operations research problems within the framework of optimal control theory."* – *Wikipedia/ optimal control*

"Optimal control is an extension of the calculus of variations, & is a mathematical optimization method for deriving control policies. The method is largely due to the work of Lev Pontryagin & Richard Bellman in the 1950s, after contributions to calculus of variations by Edward J. McShane. Optimal control can be seen as a control strategy in control theory." – Wikipedia/optimal control

General method. Optimal control deals with the problem of finding a control law for a given system s.t. a certain optimality criterion is achieved. A control problem includes a cost functional that is a function of state & control variables. An *optimal control* is a set of differential equations describing the paths of the control variables that minimize the cost function. The optimal control can be derived using Pontryagin's maximum principle (a necessary condition also known as Pontryagin's minimum principle or simply Pontryagin's principle), or by solving the Hamilton–Jacobi–Bellman equation0 (a sufficient condition).

Example 1. *Consider a car traveling in a straight line on a hilly road. Question: How should the driver press the accelerator pedal in order to minimize the total traveling time? The term* control law *refers specifically to the way in which the driver presses the accelerator & shifts the gears. The* system *consists of both the car & the road, & the* optimality criterion *is the minimization of the total traveling time. Control problems usually include ancillary constraints. E.g., the amount of available fuel might be limited, the accelerator pedal cannot be pushed through the floor of the car, speed limits, etc.*

A proper cost function will be a mathematical expression giving the traveling time as a function of the speed, geometrical considerations, & initial conditions of the system. Constraints are often interchangeable with the cost function.

Another related optimal control problem may be to find the way to drive the car so as to minimize its fuel consumption, given that it must complete a given course in a time not exceeding some amount. Yet another related control problem may be to minimize the total monetary cost of completing the trip, given assumed monetary prices for time & fuel.

An abstract framework. Minimize the continuous-time cost functional

$$J(t_0, t_\mathrm{f}, \mathbf{x}(\cdot), \mathbf{u}(\cdot)) \coloneqq E(\mathbf{x}(t_0), t_0, \mathbf{x}(t_\mathrm{f}), t_\mathrm{f}) + \int_{t_0}^{t_\mathrm{f}} F(t, \mathbf{x}(t), \mathbf{u}(t))\,\mathrm{d}t, \tag{1}$$

subject to the 1st-order dynamic constraints (the *state equation*)

$$\dot{\mathbf{x}}(t) = \mathbf{f}(t, \mathbf{x}(t), \mathbf{u}(t)), \tag{2}$$

the algebraic *path constraints*

$$\mathbf{h}(t, \mathbf{x}(t), \mathbf{u}(t)) \leq \mathbf{0}, \tag{3}$$

& the endpoint conditions

$$\mathbf{e}(t_0, \mathbf{x}(t_0), t_\mathrm{f}, \mathbf{x}(t_\mathrm{f})) = \mathbf{0}, \tag{4}$$

where $\mathbf{x}(t)$: the *state*, $\mathbf{u}(t)$ is the *control*, t: the independent variable (generally speaking, time), t_0: the initial time, & t_f: the terminal time. The terms E, F are called the *endpoint cost* & *running cost*, respectively. In the calculus of variations, E, F are referred to as the Mayer term & the Lagrangian, respectively. Furthermore, it is noted that the path constraints are in general *inequality* constraints & thus may not be active (i.e., $= 0$) at the optimal solution. It is also noted that the optimal control problem as stated above may have multiple solutions (i.e., the solution may not be unique). Thus, it is most often the case that any solution $(t_0^\star, t_\mathrm{f}^\star, \mathbf{x}^\star(t), \mathbf{u}^\star(t))$ to the optimal control problem is *locally minimizing/ minimizer*.

3.1 Linear Quadratic Control

A special case of the general nonlinear optimal control problem is the linear quadratic (LQ) optimal control problem. The LQ problem is stated as follows. Minimize the *quadratic* continuous-time cost functional

$$J = \frac{1}{2}\mathbf{x}^{\top}(t_{\mathrm{f}})S_{\mathrm{f}}\mathbf{x}(t_{\mathrm{f}}) + \frac{1}{2}\int_{t_0}^{t_{\mathrm{f}}} \mathbf{x}^{\top}(t)Q(t)\mathbf{x}(t) + \mathbf{u}^{\top}(t)R(t)\mathbf{u}(t)\,\mathrm{d}t, \tag{5}$$

subject to the linear 1st-order dynamic constraints

$$\begin{cases} \dot{\mathbf{x}}(t) = A(t)\mathbf{x}(t) + B(t)\mathbf{u}(t), \\ \mathbf{x}(t_0) = \mathbf{x}_0. \end{cases} \tag{6}$$

A particular form of the LQ problem arising in many control system problems is that of the *linear quadratic regulator* (LQR) where all of the matrices (i.e., A, B, Q, R) are constant, the initial time is arbitrarily set to 0, & the terminal time is taken in the limit $t_{\mathrm{f}} \to 0$ (this last assumption is what is known as *infinite horizon*). The LQR problem is stated as follows. Minimize the infinite horizon quadratic continuous-time cost functional

$$J = \frac{1}{2}\int_{0}^{\infty} \mathbf{x}^{\top}(t)Q\mathbf{x}(t) + \mathbf{u}^{\top}(t)R\mathbf{u}(t)\,\mathrm{d}t, \tag{7}$$

subject to the *linear time-invariant* 1st-order dynamic constraints

$$\begin{cases} \dot{\mathbf{x}}(t) = A\mathbf{x}(t) + B\mathbf{u}(t), \\ \mathbf{x}(t_0) = \mathbf{x}_0. \end{cases} \tag{8}$$

In the finite-horizon case the matrices are restricted in that Q, R are positive semi-definite & positive definite, respectively. In the infinite-horizon case, however, the matrices Q, R are not only positive-semidefinite & positive-definite, respectively, but are also constant. These additional restrictions on Q, R in the infinite-horizon case are enforced to ensure that the cost functional remains positive. Furthermore, in order to ensure that the cost function is *bounded*, the additional restriction is imposed that the pair (A, B) is *controllable*. Note that the LQ or LQR cost functional can be thought of physically as attempting to minimize the *control energy* (measured as a quadratic form).

The infinite horizon problem, i.e., LQR, may seem overly restrictive & essentially useless because it assumes that the operator is driving the system to 0-state & hence driving the output of the system to 0. This is indeed correct. However the problem of driving the output to a desired nonzero level can be solved *after* the zero output one is. In fact, can prove that this secondary LQR problem can be solved in a very straightforward manner. It has been shown in classical optimal control theory that the LQ (or LQR) optimal control has the feedback form

$$\mathbf{u}(t) = -K(t)\mathbf{x}(t), \tag{9}$$

where $K(t)$ is a properly dimensioned matrix, given as

$$K(t) = R^{-1}B^{\top}S(t), \tag{10}$$

& $S(t)$ is the solution of the differential Riccati equation. The differential Riccati equation is given as

$$\dot{S}(t) = -S(t)A - A^{\top}S(t) + S(t)BR^{-1}B^{\top}S(t) - Q. \tag{11}$$

For the finite horizon LQ problem, the Riccati equation is integrated backward in time using the terminal boundary condition

$$S(t_{\mathrm{f}}) = S_{\mathrm{f}}. \tag{12}$$

For the infinite horizon LQR problem, the differential Riccati equation is replaced with the *algebraic* Riccati equation (ARE) given as

$$-SA - A^{\top}S + SBR^{-1}B^{\top}S - Q = \mathbf{0}. \tag{13}$$

Understanding that the ARE arises from infinite horizon problem, the matrices A, B, Q, R are all constant. It is noted that there are in general multiple solutions to the algebraic Riccati equation & the *positive definite* (or positive semi-definite) solution is the one that is used to compute the feedback again. The LQ (LQR) problem was elegantly solved by Rudolf E. Kálmán.

3.2 Numerical Methods for Optimal Control

"Optimal control problems are generally nonlinear & therefore, generally do not have analytic solutions, e.g., like the linear-quadratic optimal control problem. As a result, it is necessary to employ numerical methods to solve optimal control problems. In the early years of optimal control (c. 1950s–1980s) the favored approach for solving optimal control problems was that of *indirect methods*. In an indirect method, the calculus of variations is employed to obtain the 1st-order optimality conditions. These conditions result in a 2-point (or, in the case of a complex problem, a multi-point) BVP. This BVP actually has a special

structure because it arises from taking the derivative of a Hamiltonian. Thus, the resulting dynamical system is a Hamiltonian system of the form

$$\begin{cases} \dot{\mathbf{x}} = \partial_{\boldsymbol{\lambda}} H, \\ \dot{\boldsymbol{\lambda}} = -\partial_{\mathbf{x}} H, \end{cases} \tag{14}$$

where $H = F + \boldsymbol{\lambda}^\top \mathbf{f} - \boldsymbol{\mu}^\top \mathbf{h}$ is the *augmented Hamiltonian* & in an indirect method, the BVP is solved (using the appropriate boundary or *transversality* conditions). The beauty of using an indirect method is that the state & adjoint, i.e., $\boldsymbol{\lambda}$, are solved for & the resulting solution is readily verified to be an extremal trajectory. The disadvantage of indirect methods is that the BVP is often extremely difficult to solve (particularly for problems that span large time intervals or problems with interior point constraints). A well-known software program that implements direct methods is BNDSCO.

The approach that has risen to prominence in numerical optimal control since the 1980s is that of so-called *direct methods*. In a direct method, the state or the control, or both, are approximated using an appropriate function approximation (e.g., polynomial approximation or piecewise constant parameterization). Simultaneously, the cost functional is approximated as a *cost function*. Then, the coefficients of the function approximations are treated as optimization variables & the problem is "transcribed" to a nonlinear optimization problem of the form

$$\min F(\mathbf{z}) \tag{15}$$

subject to the algebraic constraints

$$\mathbf{g}(\mathbf{z}) = \mathbf{0}, \ \mathbf{h}(\mathbf{z}) \leq \mathbf{0}. \tag{16}$$

Depending upon the type of direct method employed, the size of the nonlinear optimization problem can be quite small (e.g., as in a direct shooting or quasilinearization method), moderate (e.g., pseudospectral optimal control) or may be quite large (e.g., a direct collocation method). In the latter case (i.e., a collocation method), the nonlinear optimization problem may be literally thousands $a \cdot 10^3$ to tens of thousands $a \cdot 10^4$ of variables & constraints. Given the size of many NLPs arising from a direct method, it may appear somewhat counter-intuitive that solving the nonlinear optimization problem is easier than solving the BVP. It is, however, the fact that the NLP is easier to solve than the BVP. The reason for the relative ease of computation, particularly of a direct collocation method, is that the NLP is *sparse* & many well-known software programs exist (e.g., SNOPT) to solve large sparse NLPs. As a result, the range of problems that can be solved via direct methods (particularly direct *collocation methods* which are very popular these days) is significantly larger than the range of problems that can be solved via indirect methods. In fact, direct methods have become so popular these days that many people have written elaborate software programs employing these methods. In particular, many such programs include DIRCOL, SOCS, OTIS, GESOP/ASTOS, DITAN., & PyGMO/PyKEP. In recent years, due to the advent of the MATLAB programming language, optimal control software in MATLAB has become more common. Examples of academically developed MATLAB software tools implementing direct methods include RIOTs, DIDO, DIRECT, FALCON.m, & GPOPs, while an example of an industry developed MATLAB tool is PROPT. These software tools have increased significantly the opportunity for people to explore complex optimal control problems both for academic research & industrial problems. Finally, it is noted that general-purpose MATLAB optimization environments such as TOMLAB have made coding complex optimal control problems significantly easier than was previously possible in languages such as C & FORTRAN.

3.3 Discrete-Time Optimal Control

"The examples thus far have shown continuous time systems & control solutions. In fact, as optimal control solutions are now often implemented digitally, contemporary control theory is now primarily concerned with discrete time systems & solutions. The theory of Consistent Approximations provides conditions under which solutions to a series of increasingly accurate discretized optimal control problem converge to the solution of the original, continuous-time problem. Not all discretization methods have this property, even seemingly obvious ones. E.g., using a variable step-size routine to integrate the problem's dynamic equations may generate a gradient which does not converge to 0 (or point in the right direction) as the solution is approached. The direct method RIOTS is based on the Theory of Consistent Approximation.

A common solution strategy in many optimal control problems is to solve for the costate (sometimes called the shadow price) $\lambda(t)$. The costate summaries in 1 number the marginal value of expanding or contracting the state variable next turn. The marginal value is not only the gains accruing to it next turn but associated with the duration of the program. It is nice when $\lambda(t)$ can be solved analytically, but usually, the most one can do is describe it sufficiently well that the intuition can grasp the character of the solution & an equation solver can solve numerically for the values.

Having obtained $\lambda(t)$, the turn-t optimal value for the control can usually be solved as a differential equation conditional on knowledge of $\lambda(t)$. Again it is infrequent, especially in continuous-time problems, that one obtains the value of the control or the state explicitly. Usually, the strategy is to solve for thresholds & regions that characterize the optimal control & use a numerical solver to isolate the actual choice values in time.

Example 2. *"Consider the problem of a mine owner who must decide at what rate to extract ore from their mine. They own rights to the ore from date 0 to date T. At date 0 there is x_0 ore in the ground, & the time-dependent amount of ore $x(t)$ left in the ground declines at the rate of $u(t)$ that the mine owner extracts it. The mine owner extracts ore at cost $\frac{u(t)^2}{x(t)}$ (the cost of extraction increasing with the square of the extraction speed & the inverse of the amount of ore left) & sells ore at a constant price p. Any ore left in the ground at time T cannot be sold & has no value (there is no "scrap value"). The owner chooses the rate of extraction varying with time $u(t)$ to maximize profits over the period of ownership with no time discounting. See discrete-time version vs. Continuous-time version." – Wikipedia/optimal control/finite time*

3.4 Optimal Control for PDEs – Điều Khiển Tối Ưu Cho Phương Trình Vi Phân Đạo Hàm Riêng

Resources – Tài nguyên.

1. [Lio71]. Jacques-Louis Lions. *Optimal control of systems governed by PDEs.*
2. [Trö10]. Fredi Tröltzsch. *Optimal Control of PDEs.*

3.4.1 Optimal Control for Navier–Stokes Equations – Điều Khiển Tối Ưu Cho Phương Trình Navier–Stokes

4 Shape Optimization – Tối Ưu Hình Dạng

Resources – Tài nguyên.

1. [AH01]. Grégoire Allaire, Antoine Henrot. *On some recent advances in shape optimization.*
2. [Aze20]. Hideyuki Azegami. *Shape Optimization Problems.*
3. [BW23]. Catherine Bandle, Alfred Wagner. *Shape Optimization: Variations of Domains & Applications.*
4. [DZ01; DZ11]. Michael C. Delfour, Jean-Paul Zolésio. *Shapes & Geometries.*
5. [HM03]. J. Haslinger, R. A. E. Mäkinen. *Introduction to Shape Optimization.*
6. [MP10]. Bijan Mohammadi, Olivier Pironneau. *Applied Shape Optimization for Fluids.*
7. [MZ06]. Marwan Moubachir, Jean-Paul Zolésio. *Moving Shape Analysis & Control.*
8. Stephan Schmidt. Master course: *Shape & Geometry.* Humboldt University of Berlin. [written in German, taught in English & German].
9. [SZ92]. Jan Sokołowski, Jean-Paul Zolésio. *Introduction to Shape Optimization.*
10. [Wal15]. Shawn W. Walker. *The Shapes of Things.*

    Differential equations on surfaces. Differential geometry is useful for understanding mathematical models containing geometric PDEs, e.g., surface/manifold version of the standard Laplace equation, which requires the development of the surface gradient & surface Laplacian operators – the usual gradient ∇ & Laplacian $\Delta = \nabla \cdot \nabla$ operators defined on a surface (manifold) instead of standard Euclidean space $\mathbb{R}^n$. *Advantage*: provide alternative formulas for geometric quantities, e.g., the summed (mean) curvature, that are much clearer than the usual presentation of texts on differential geometry.

    Differentiating w.r.t. Shape. The approach to differential geometry is advantageous for developing the framework of *shape differential calculus* – the study of how quantities change w.r.t. changes of independent "shape variable".

    "The framework of shape differential calculus provides the tools for developing the equations of mean curvature flow & Willmore flow, which are geometric flows occurring in many applications such as fluid dynamics & biology." – [Wal15, p. 2]

    The shape perturbation $\delta J(\Omega; V)$ is similar to the gradient operator, which is a directional derivative, analogous to $V \cdot \nabla f$ where V is a given direction, providing information about the local slope, or the sensitivity of a quantity w.r.t. some parameters.

    It takes only 2 or 3 numbers to specify a point (x, y) in 2D & a point (x, y, z) in 3D, whereas an "infinite" number of coordinate pairs is needed to specify a domain Ω. V is a 2D/3D vector in the scalar function setting; for a shape functional, V is a full-blown function requiring definition at every point in Ω. This "infinite dimensionality" is the reason for using the notation $\delta J(\Omega; V)$ to denote a shape perturbation. $\delta J(\Omega; V)$ indicates how we should change Ω to decrease J, similarly to how $\nabla f(x, y)$ indicates how the coordinate pair (x, y) should change to decrease f, which opens up the world of shape optimization.

    3 schools of shape optimization. Cf. engineering shape optimization vs. applied shape optimization [MP10] vs. theoretical shape optimization [SZ92; DZ11].

    Shape perturbations allow us to "climb down the hill" in the infinite dimensional setting of shape, which is a powerful tool for producing sophisticated engineering designs in an automatic way.

    Extrinsic vs. intrinsic point of views. To make the discussion as clear as possible, we adopt the *extrinsic* point of view: curves & surfaces are assumed to lie in a Euclidean space of higher dimension. The ambient space is 3D Euclidean space. Alternatively, there is the *intrinsic* point of view, i.e., the surface is not assumed to lie in an ambient space, i.e., one is not allowed to reference anything "outside" of the surface when defining it. Moreover, no mathematical structures "outside" of the surface can be utilized. Walker [Wal15] did not adopt the intrinsic view or consider higher dimensional manifolds, general embedding dimensions, etc. for the reasons: [Wal15] is meant as a *practical guide* to differential geometry & shape differentiation that can be used by researchers in other fields.

    - [Wal15] is meant to be used as background information for deriving physical models where geometry plays a critical role. Because most physical problems of interest take place in 3D Euclidean space, the extrinsic viewpoint is sufficient.

- Many of the proofs & derivations of differential geometry relations simplify dramatically for 2D surfaces in 3D & require only basic multivariable calculus & linear algebra.
- The concepts of *normal vectors & curvature* are harder to motivate with the intrinsic viewpoint. *What does it meant for a surface to "curve through space" if you cannot talk about the ambient space?*
- Walker wants to keep in mind applications of this machinery to geometric PDEs, fluid dynamics, numerical analysis, optimization, etc. An interesting application of this methodology is for the development of numerical methods for mean curvature flow & surface tension driven fluid flow. Ergo (= therefore), the extrinsic viewpoint is often more convenient for computational purposes.
- Walker wants his framework to be useful for analyzing & solving *shape optimization* problems, i.e., optimization problems where geometry/shape is the control variable.

Prerequisites. "When reading any mathematical text, the reader must have a certain level of mathematical "maturity" in order to efficiently learn what is in the text.

Example 3 ([Wal15], Sect. 1.2.1, pp. 1–2)**.** *Let $f = f(r,\theta)$ be a smooth function defined on the disk $B_{R,2}(0,0)$ of radius R in terms of polar coordinates. The integral of f over $B_{2,R}(0,0)$ $J \coloneqq \int_{B_{2,R}(0,0)} f \,\mathrm{d}\mathbf{x} = \int_0^{2\pi}\int_0^R f(r,\theta)\,\mathrm{d}r\,\mathrm{d}\theta$ depends on R. Assume f also depends on R, i.e., $f = f(r,\theta,R)$ with a physical example: J is the* net flow rate *of liquid through a pipe with cross-section Ω, then f is the flow rate per unit area & could be the solution of a PDE defined on Ω, e.g., a Navier–Stokes fluid flowing in a circular pipe. Advantageous to know the* sensitivity *of J w.r.t. R, e.g., for optimization purposes. Differentiate J w.r.t. R:*

$$\frac{d}{dR}J = \int_0^{2\pi}\left(\frac{d}{dR}\int_0^R f(r,\theta;R)r\,\mathrm{d}r\right)\mathrm{d}\theta = \int_0^{2\pi}\int_0^R f'(r,\theta;R)r\,\mathrm{d}r\,\mathrm{d}\theta + \int_0^{2\pi} f(R,\theta;R)\,\mathrm{d}\theta.$$

The dependence of f on R can more generally be viewed as dependence on $B_{R,2}(0,0)$, i.e., $f(\cdot;R) \equiv f(\cdot;B_{R,2}(0,0))$. Rewriting d/dRJ using Cartesian coordinates $\mathbf{x}$:

$$\frac{d}{dR}J = \int_{B_{R,2}(0,0)} f'(\mathbf{x};\Omega)\,\mathrm{d}\mathbf{x} + \int_{S_{R,2}(0,0)} f(\mathbf{x};\Omega)\,\mathrm{d}S(\mathbf{x}), \tag{17}$$

where $\mathrm{d}\mathbf{x}$ is the volume measure, $\mathrm{d}S(\mathbf{x})$ is the surface area measure.

Example 4 (Surface height function of a hill)**.** *Let $f = f(x,y)$ be a function describing the surface height of the hill, where (x,y) are the coordinates of our position. Then, by using basic multivariate calculus, finding a direction that will move us downhill is equivalent to computing the gradient (vector) of f & moving in the opposite direction to the gradient. In this sense, we do not need to "see" the whole function. We just need to* locally *compute the gradient ∇f, analogous to feeling the ground beneath.*

Example 5 (Engineering shape optimization: minimizing drag with Navier–Stokes flow of fluid past a rigid body, [Wal15], pp. 3–5)**.** *A shape functional representing the drag:*

$$J_{\mathrm{d}}(\Omega) \coloneqq -\mathbf{u}_{\mathrm{out}} \cdot \int_{\Gamma_0} \boldsymbol{\sigma}(\mathbf{u},p)\mathbf{n}\,\mathrm{d}\Gamma = \frac{2}{\mathrm{Re}}\int_\Omega |\boldsymbol{\varepsilon}(\mathbf{u})|^2\,\mathrm{d}\mathbf{x} \geq 0, \tag{18}$$

which physically represents the net force that must be applied to Ω_{B} to keep it stationary while being acted upon by the imposed flow field & represents the total amount of viscous dissipation of energy (per unit of time) in the fluid domain Ω. Using the machinery of shape perturbations, $\delta J_{\mathrm{d}}(\Omega;V)$ indicates how J_{d} changes when we perturb Ω in the direction V. Hence, we can use this information to change Ω in small steps so as to slowly deform Ω into a shape that has better (lower) drag characteristics. A numerical computation: 2 large vortices appear behind the body, which indicate a large amount of viscous dissipation, i.e., large drag. The optimization process then computes $\delta J_{\mathrm{d}}(\Omega^0;V)$ for many different choices of V & chooses the one that drives down J_{d} the most. This choice of V is used to deform Γ_{B}^0 into a new shape Γ_{B}^1 at iteration 1, with only a small difference between Γ_{B}^0 & Γ_{B}^1. This process is repeated many times. Note how the vortices are eliminated by the more slender shape.

- Condition (V)***
- Family of transformations $\{T_s : 0 \leq s \leq \tau\}$.
- Perturbed domain $\Omega_s = \Omega_s(V) = T_s(V)(\Omega)$.
- Assume that the velocity field V satisfies V_D and, in addition, $V \in C^0([0,\tau];C^1_{\mathrm{loc}}(\mathbb{R}^d,\mathbb{R}^d))$ and $\tau > 0$ is small enough such that the Jacobian J_s is strictly positive: $J_s(X) \coloneqq \det DT_s(X) > 0$, $\forall s \in [0,\tau]$, where $DT_s(X)$ is the *Jacobian matrix* of the transformation $T_s = T_s(V)$ associated with the velocity vector field V.

4.1 Domain Integrals

- Given $\varphi \in W^{1,1}_{\mathrm{loc}}(\mathbb{R}^d)$, consider for $s \in [0,\tau]$ the volume integral

$$J(\Omega_s(V)) \coloneqq \int_{\Omega_s(V)} \varphi\,\mathrm{d}\mathbf{x} = \int_\Omega \varphi \circ T_s J_s\,\mathrm{d}\mathbf{x}. \tag{19}$$

$$dJ(\Omega; V) = \frac{d}{ds} J(\Omega_s(V))|_{s=0} = \int_\Omega \nabla\varphi \cdot V(0) + \varphi \operatorname{div} V(0)\,\mathrm{d}\mathbf{x} = \int_\Omega \operatorname{div}(\varphi V(0))\,\mathrm{d}\mathbf{x}. \tag{20}$$

If Ω has a Lipschitzian boundary, by Stokes's theorem:

$$dJ(\Omega; V) = \int_\Gamma \varphi V(0) \cdot \mathbf{n}\,\mathrm{d}\Gamma. \tag{21}$$

Theorem 1 ([DZ11], Thm. 4.1, pp. 482–483)**.** *Let $\varphi \in W^{1,1}_{\mathrm{loc}}(\mathbb{R}^d)$. Assume that the vector field $V = \{V(s) : 0 \le s \le \tau\}$ satisfies condition (V).*

(i) For each $s \in [0,\tau]$ the map $\varphi \mapsto \varphi \circ T_s : W^{1,1}_{\mathrm{loc}}(\mathbb{R}^d) \to W^{1,1}_{\mathrm{loc}}(\mathbb{R}^d)$ and its inverse are both locally Lipschitzian and $\nabla(\varphi \circ T_s) = {}^*DT_s \nabla\varphi \circ T_s$.

(ii) If $V \in C^0([0,\tau]; C^1_{\mathrm{loc}}(\mathbb{R}^d, \mathbb{R}^d))$, then the map $s \mapsto J_s : [0,\tau] \to C^0_{\mathrm{loc}}(\mathbb{R}^d)$ is differentiable and

$$\frac{dJ_s}{ds} = [\nabla \cdot V(s)] \circ T_s J_s \in C^0_{\mathrm{loc}}(\mathbb{R}^d). \tag{22}$$

Hence the map $s \mapsto J_s$ belongs to $C^1([0,\tau]; C^0_{\mathrm{loc}}(\mathbb{R}^d))$.

$$\frac{d}{ds} DT_s(X) = DV(s, T_s(X)) DT_s(X),\ DT_0(X) = I, \tag{23}$$

$$\frac{d}{ds} \det DT_s(X) = \operatorname{tr} DV(s, T_s(X)) \det DT_s(X) = \nabla \cdot V(s, T_s(X)) \det DT_s(X),\ \det DT_0(X) = 1. \tag{24}$$

Theorem 2 ([DZ11], Thm. 4.2, pp. 483–484)**.** *Assume that there exists $\tau > 0$ such that the velocity field $V(t)$ satisfies conditions (V) and $V \in C^0([0,\tau]; C^1_{\mathrm{loc}}(\mathbb{R}^d, \mathbb{R}^d))$. Given a function $\varphi \in C(0,\tau; W^{1,1}_{\mathrm{loc}}(\mathbb{R}^d)) \cap C^1(0,\tau; L^1_{\mathrm{loc}}(\mathbb{R}^d))$ and a bounded measurable domain Ω with boundary Γ, the semiderivative of the function $J_V(s) := \int_{\Omega_s(V)} \varphi(s)\,\mathrm{d}\mathbf{x}$ at $s = 0$ is given by*

$$dJ_V(0) = \int_\Omega \varphi'(0) + \operatorname{div}(\varphi(0)V(0))\,\mathrm{d}\mathbf{x}, \tag{25}$$

where $\varphi(0)(\mathbf{x}) := \varphi(0,\mathbf{x})$ and $\varphi'(0)(\mathbf{x}) := \partial_t\varphi(0,\mathbf{x})$. In addition, Ω is an open domain with a Lipschitzian boundary Γ, then

$$dJ_V(0) = \int_\Omega \varphi'(0)\,\mathrm{d}\mathbf{x} + \int_\Gamma \varphi(0)V(0) \cdot \mathbf{n}\,\mathrm{d}\mathbf{x}. \tag{26}$$

4.2 Boundary Integrals

4.3 Material derivatives

Let $\Omega \subset D$ be a bounded domain,

5 Topology Optimization – Tối Ưu Tôpô

6 Miscellaneous

Tài liệu

[AH01] Grégoire Allaire and Antoine Henrot. "On some recent advances in shape optimization". In: *C. R. Acad. Sci. Paris* t. 329, Série II b (2001), pp. 383–396.

[Aze20] Hideyuki Azegami. *Shape optimization problems*. Vol. 164. Springer Optimization and Its Applications. Springer, Singapore, 2020, pp. xxiii+646. ISBN: 978-981-15-7618-8; 978-981-15-7617-1. DOI: 10.1007/978-981-15-7618-8. URL: https://doi.org/10.1007/978-981-15-7618-8.

[BW23] Catherine Bandle and Alfred Wagner. *Shape Optimization: Variations of Domains and Applications*. Vol. 42. De Gruyter Series in Nonlinear Analysis and Applications. De Gruyter, 2023, pp. xi+278. DOI: 10.1515/9783111025438-201. URL: https://doi.org/10.1515/9783111025438-201.

[DZ01] M. C. Delfour and J.-P. Zolésio. *Shapes and geometries*. Vol. 4. Advances in Design and Control. Analysis, differential calculus, and optimization. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2001, pp. xviii+482. ISBN: 0-89871-489-3.

[DZ11] M. C. Delfour and Jean-Paul Zolésio. *Shapes and geometries*. Second. Vol. 22. Advances in Design and Control. Metrics, analysis, differential calculus, and optimization. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2011, pp. xxiv+622. ISBN: 978-0-898719-36-9. DOI: 10.1137/1.9780898719826. URL: https://doi.org/10.1137/1.9780898719826.

[HM03] J. Haslinger and R. A. E. Mäkinen. *Introduction to shape optimization*. Vol. 7. Advances in Design and Control. Theory, approximation, and computation. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2003, pp. xviii+273. ISBN: 0-89871-536-9. DOI: 10.1137/1.9780898718690. URL: https://doi.org/10.1137/1.9780898718690.

[Lio69] J.-L. Lions. *Quelques méthodes de résolution des problèmes aux limites non linéaires*. Dunod, Paris; Gauthier-Villars, Paris, 1969, pp. xx+554.

[Lio71] J.-L. Lions. *Optimal control of systems governed by partial differential equations*. Die Grundlehren der mathematischen Wissenschaften, Band 170. Translated from the French by S. K. Mitter. Springer-Verlag, New York-Berlin, 1971, pp. xi+396.

[MP10] Bijan Mohammadi and Olivier Pironneau. *Applied shape optimization for fluids*. Second. Numerical Mathematics and Scientific Computation. Oxford University Press, Oxford, 2010, pp. xiv+277. ISBN: 978-0-19-954690-9.

[MZ06] Marwan Moubachir and Jean-Paul Zolésio. *Moving shape analysis and control*. Vol. 277. Pure and Applied Mathematics (Boca Raton). Applications to fluid structure interactions. Chapman & Hall/CRC, Boca Raton, FL, 2006, pp. xx+291. ISBN: 978-1-58488-611-2; 1-58488-611-0. DOI: 10.1201/9781420003246. URL: https://doi.org/10.1201/9781420003246.

[SZ92] Jan Sokołowski and Jean-Paul Zolésio. *Introduction to shape optimization*. Vol. 16. Springer Series in Computational Mathematics. Shape sensitivity analysis. Springer-Verlag, Berlin, 1992, pp. ii+250. ISBN: 3-540-54177-2. DOI: 10.1007/978-3-642-58106-9. URL: https://doi.org/10.1007/978-3-642-58106-9.

[Thá+25] Đỗ Đức Thái, Phạm Xuân Chung, Nguyễn Sơn Hà, Nguyễn Thị Phương Loan, Phạm Sỹ Nam, and Phạm Minh Phương. *Chuyên Đề Học Tập Toán 12*. Nhà Xuất Bản Đại Học Sư Phạm, 2025, p. 75.

[Trö10] Fredi Tröltzsch. *Optimal control of partial differential equations*. Vol. 112. Graduate Studies in Mathematics. Theory, methods and applications, Translated from the 2005 German original by Jürgen Sprekels. American Mathematical Society, Providence, RI, 2010, pp. xvi+399. ISBN: 978-0-8218-4904-0. DOI: 10.1090/gsm/112. URL: https://doi.org/10.1090/gsm/112.

[Wal15] Shawn W. Walker. *The shapes of things*. Vol. 28. Advances in Design and Control. A practical guide to differential geometry and the shape derivative. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2015, pp. ix+154. ISBN: 978-1-611973-95-2. DOI: 10.1137/1.9781611973969.ch1. URL: https://doi.org/10.1137/1.9781611973969.ch1.