Computer Aided Geometric Design 59 (2018) 49–67

Contents lists available at ScienceDirect

Computer Aided Geometric Design

www.elsevier.com/locate/cagd

Robust generation of constrained B-spline curves based on automatic differentiation and fairness optimization ☆

Lothar Birk *, T. Luke McCulloch

School of Naval Architecture and Marine Engineering, University of New Orleans, 2000 Lakeshore Drive, New Orleans, LA 70148, USA

A R T I C L E   I N F O

Article history:
Received 18 July 2016
Received in revised form 2 October 2017
Accepted 30 November 2017
Available online 8 December 2017

Keywords:
Form parameter design
Fairness optimization
Automatic differentiation
B-spline
Python

A B S T R A C T

This paper details the use of automatic differentiation in form parameter driven curve design by constrained optimization. Computer aided design, computer aided engineering (CAD/CAE), and particularly computer aided ship hull design (CASHD) are typically implemented as interactive processes in which the user obtains desired shapes by manipulation of control vertices. A fair amount of trial and error is needed to achieve the desired properties. In the variational form parameter approach taken here, the system computes vertices so that the resulting curve meets the specifications and is optimized with respect to a fairness criteria. Implementation of curve design as an optimization problem requires extensive derivative calculations. The paper illustrates how the programming burden can be eased through the use of automatic differentiation techniques. A variational curve design framework has been implemented in Python, and applications to CASHD curve design are shown. The new method is robust and allows great flexibility in the selection of constraints. Offsets, tangents, and curvature may be imposed anywhere along the curve. Form parameters may also be used to define straight segments within a curve, require the curve to enclose specified forms, or specify relationships between curve properties.

1. Introduction

Current CAD/CAE systems have developed to the point where they not only serve as replacement for the drawing board but also provide complete product documentation for the whole life cycle of the system. A next logical step in the development of CAD and CAE methods is their comprehensive integration with engineering analysis and exploiting results in product optimization and design space exploration. One of the most promising product development methods for computer aided design of complex hydrodynamic or aerodynamic hulls like ships, cars, and aircraft is so called “form parameter design.”

In traditional, interactive hull design, the user places and manipulates vertices of a control polygon or mesh to create a B-spline/NURBS curve or surface. The shape is subsequently analyzed and its form parameters, i.e. its properties, are computed as output. Form parameters of interest may be offsets, tangents, enclosed areas, centroids, and more. Typically, a design goes through many iterations before the desired properties are achieved. In addition, considerable rework may be required to fair interactively designed curves and surfaces. Form parameter design reverses this process and is therefore also

☆ This paper has been recommended for acceptance by Thomas A. Grandine.

* Corresponding author.
E-mail addresses: lbirk@uno.edu (L. Birk), tmccullo@uno.edu (T.L. McCulloch).

https://doi.org/10.1016/j.cagd.2017.11.005
0167-8396/

known as "inverse design." The properties are used as input and the software returns the desired shape in the form of a control polygon or control mesh. Fairness is built into the form parameter design, thus eliminating time consuming rework.

Form parameter design may be conveniently implemented as an optimization problem. The vertex locations serve as free variables and the form parameters are implemented as a set of equality and inequality constraints. Fairness criteria form the basis of the objective function which quantifies the quality of a curve design. In ship design, fair curves and surfaces usually imply less resistance and lower manufacturing cost. In general, the inverse design problem is nonlinear. For an introduction to nonlinear programming and shape design within the context of marine applications see Birk and Harries (2003).

Research in the field of form parameter driven curve and shape design goes back many years. The reader interested in curve design for ships should see Nowacki and Reed (1974) and Rogers (1977) for landmark developments in the computer-aided design of complex hull forms. A general overview of computational curve design as it has progressed over the previous decade is given in Nowacki (2010). The references Nowacki et al. (1995), Harries and Nowacki (1999), Bole (1997), Bole and Lee (2006), Harries (1998), Kim (2004) present methods for the design of complex, fairness optimized ship hulls. The form parameter design method presented in this paper builds on the work by Harries (1998) and significantly extends the range of applicable form parameters.

In naval architecture practice a curve is considered fair, if its curvature distribution is smooth, and the curve features only the necessary changes in the direction of curvature. The quality of a curve is quantified with a "fairness functional" which is minimized during the optimization (see Section 2.2). The fairness functional is augmented with equality and inequality constraints and the minimum is found by the method of Lagrange multipliers. The constraints represent desired form parameter values. The optimization problem is solved here using Lagrange's method of multipliers. For background on the Lagrange multiplier technique in optimization, a multitude of sources are available, such as Christodoulos and Panos (2009), Luenberger and Ye (2008), Boyd and Vandenberghe (2004).

In principle, any nonlinear programming algorithm may be used to minimize the augmented fairness functional. However, the form parameter design method here is developed as the basis for future research in hull shape optimization. For that reason, control of interfaces and interoperability of the components was more important than tweaking numerical efficiency. The authors choose to stick with the Lagrangian multiplier technique coupled with a simple Newton–Raphson solver because it is easy to implement and has proven stable enough for the envisioned purpose.

The necessary condition for a minimum of the augmented Lagrangian function commonly results in a system of nonlinear equations whose solution requires extensive derivative calculations. In this paper the computation of derivatives of the Lagrangian is handled by automatic differentiation (AD). The use of AD simplifies the implementation of an augmented Lagrangian and enables the inclusion of complicated form parameters, like for example, the position of cusps, straight segments within the curve, or designing around an interior shape.

The technical development of automatic differentiation (AD) is described in Griewank (1989). Christianson (2008) reviews the performance of "forward mode" versus "reverse mode" AD. Applications can be found in Arbenz and Gander (1986), Eberhard and Bischof (1999), Giles (2001), Giles et al. (2005), Hovland et al. (1998), Lin et al. (2013). A detailed review of the AD scheme employed in this paper is found in Neidinger (2010) which presents schemes are for first and higher order differentiation. The presented inverse design procedure is implemented in the interpreted programming language Python (Rossum, 1995). AD packages for Python are readily available – see `pyadolc` (Walter, 2008), `algopy` (Walter, 2013), `pycppad` (Bell and Walter, 2012), `CasADI` (Andersson et al., 2012), and `ad` (Lee, 2008) – The Python optimization package `Theano` also supports automatic differentiation (Theano Development Team, 2016).We choose to implement our own AD version to retain control of the interface and maximize interoperability with the curve design utilities. In our application, only first and second order derivatives are needed which simplifies AD implementation.

B-splines are employed as the mathematical basis for curve design. Development of B-splines has been extensively covered in the technical literature, see de Boor (1978), Ramshaw (1987). An extensive reference about definition, properties, and implementation of algorithms for the manipulation of B-spline curves is provided in the well known book by Piegl and Tiller (1997). Only the vertices are used as free variables in the optimization. All weights are preset by the user and remain unchanged. Discarding the weights as a free variable increases the robustness of our procedure. We also restrict ourselves to uniform knot vectors for open curves. This is rather a choice of convenience than a requirement. The order of a curve and its knot vector define the set of basis functions. Adapting the knot vector to the geometry of a problem is useful to attain a better curve parameterization. However, it would require re-computation of the basis functions whenever the knot vector changes. Keeping the knot vector unchanged reduces the numerical effort during the fairness optimization because the values of basis functions and their derivatives can be precomputed and stored for repeated use.

The following sections are organized as follows: In Section 2 we recapture the B-spline curve design algorithm as introduced by Harries (1998). Although we use the same concepts, our implementation is based on automatic differentiation, which makes it easier to implement new sets of form parameters. The basics of automated differentiation (AD) are summarized in Section 3.

The following Section 4 focuses on the actual curve design procedure. It combines AD with Lagrangian optimization into an algorithm that is effective and robust. This section will elucidate the details necessary for AD to be used with B-splines, including the computation of B-spline properties, and nonlinear optimization. Finally, Section 5 employs the presented methodology to the form parameter driven design of constraint curves. Special emphasis is put on problems that are useful in parametric ship hull form design. The conclusions highlight advantages of automated differentiation in curve design and discuss possibilities for further development.