A Symbolic-Numeric Approach to Tube Modeling in CAD Systems

Gerrit Sobottka and Andreas Weber

Institut für Informatik II, Universität Bonn, Römerstr. 164, Bonn, Germany
E-mail: {sobottka,weber}@cs.uni-bonn.de

Abstract. *In this note we present a symbolic-numeric method to the problem of tube modeling in CAD systems. Our approach is based on the Kirchhoff kinetic analogy which allows us to find analytic solutions to the static Kirchhoff equations for rods under given boundary conditions.*
Keywords: Kirchhoff rod; boundary value problems; automatic differentiation

1 Introduction

In this short note we address the problem of physics based tube modeling which frequently occurs in *computer aided design* (CAD) applications. The task at hand is to connect a tube of given length and with certain material properties to connectors fixed in space. In particular, we aim to predict the internal forces and torques along the tube for dimensioning purposes as well as the final configuration. The equilibrium shape of the tube is governed by the well known static *Kirchhoff equations*. Along with the boundary conditions at both ends defined by the position and orientation of the connectors we have to solve a two-point boundary value problem (BVP). Such BVPs can be solved employing standard shooting techniques which usually perform at slow convergence rates. Our approach is based on the analytic solution to the static Kirchhoff equations and is a continuation of our work described in [1], to which we refer for more details on the basis of the method. In addition to adding a new case of boundary conditions we will also show how the "symbolic-numeric method" introduced in [1] can be made "more symbolic" by using the Jacobian Matrix in symbolic form within the numerical part.

2 Related Work

While the number of publications on solution methods for Kirchhoff equations is large, the task of tube modeling based on these equations has rarely been addressed before. Grégoire and Schömer [2] use an extended spring-mass system that is solved with an energy minimizing algorithm. In [3] Healey and Metha present a method to solve associated BVPs by augmenting the system of boundary conditions by a constraint on the magnitude of the quaternions used for the parameterization of the rotations. In particular, they show that if these constraints are met at the end points they are also met on the whole domain. In [4] a geometrically exact approach is proposed, which is based on the explicit solution of the kinematic relation based on Rodriguez' formula. Henderson and Neukirch [5] study spatial equilibria of clamped elastica based on Kirchhoff rods. In contrast to their work we do not restrict ourselves to the case where the tangents at the end points are collinear. In [6] Nizette and Goriely study explicit solution of the static Kirchhoff equations in terms of Euler-Kirchhoff filaments.

3 Physics Based Tube Modeling

3.1 The symbolic part

Let $\mathbf{r}(s) : [a_1, a_2] \in \mathbb{R} \mapsto \mathbb{R}^3$ be the centerline of the tube. The centerline is furnished with a set of right-handed orthonormal triads $\{\mathbf{d}_1, \mathbf{d}_2, \mathbf{d}_3\}$, such that $\mathbf{d}_3 = \mathbf{r}'$ is the tangent of the centerline and $\mathbf{d}_1, \mathbf{d}_2$ span the cross section plane at each point of the rod. Further, we assume the tube to be inextensible, unshearable, and initially straight. The equilibrium state is given by the static Kirchhoff equations:

$$\mathbf{F}' = \mathbf{0}, \tag{1}$$
$$\mathbf{M}' + \mathbf{d}_3 \times \mathbf{F} = \mathbf{0}, \tag{2}$$
$$\mathbf{M} = u_1 \cdot \mathbf{d}_1 + u_2 \cdot \mathbf{d}_2 + b \cdot u_3 \cdot \mathbf{d}_3, \tag{3}$$

where $\mathbf{F}$ and $\mathbf{M}$ are the internal force and the torque of the rod. Note that since we assume the tube to have a circular cross section we use the scaled form of the constitutive equation for $\mathbf{M}$, where $b = 1/(1+\nu)$ with ν being Poisson's ratio.

Further, we have the kinematic relation $\mathbf{d}_i = \mathbf{u} \times \mathbf{d}_i$, where $\mathbf{u} = \{u_1, u_2, u_3\}$ is the twist vector. The components of the twist vector as well as the local directors $\{\mathbf{d}_i\}$ are conveniently expressed in terms of Euler angles (φ, θ, ψ) w.r.t. to the global frame $\{\mathbf{e}_x, \mathbf{e}_y, \mathbf{e}_z\}$. With

$$\mathbf{F} = F \cdot \mathbf{e}_z, \tag{4}$$
$$M_z' = \mathbf{M}_z' \cdot \mathbf{e}_z = 0, \tag{5}$$
$$M_3' = \mathbf{M}_3' \cdot \mathbf{d}_3 = 0, \tag{6}$$
$$H = \frac{1}{2} \cdot \mathbf{M} \cdot \mathbf{u} + \mathbf{F} \cdot \mathbf{d}_3, \tag{7}$$

being first integrals, which do not depend on the arc length parameter, we obtain the following equations for the Euler angles:

$$\varphi' = \frac{M_z - M_3 \cdot z}{1 - z^2}, \tag{8}$$
$$\psi' = \frac{M_3 - M_z \cdot z}{1 - z^2} + M_3 \cdot \left(\frac{1}{b} - 1\right), \tag{9}$$
$$z'^2 = 2F \cdot (h - z) \cdot \left(1 - z^2\right) - (M_z - M_3 \cdot z)^2, \tag{10}$$

where $z = \cos\theta$ and $h = 1/F \cdot \left(H - M_3^2/2b\right)$. The right hand side of z'^2 is a cubic polynomial in z with the roots z_1, z_2, z_3, such that $-1 \leq z_1 \leq z_2 \leq 1 \leq z_3$. The solution is given by

$$z = z_1 + (z_2 - z_1) \cdot \mathrm{sn}^2 \left[\lambda(s + s_0), k\right], \tag{11}$$

where $\lambda = \sqrt{1/2 \cdot F \cdot (z_3 - z_1)}$, $k = \sqrt{(z_2 - z_1)/(z_3 - z_1)}$, and sn is one of the Jacobi elliptic function. With the function z at hand the solutions for φ and ψ are obtained by directly integrating the above equations [1]:

$$\varphi = \int_0^s \frac{M_z - M_3 \cdot z(\sigma)}{1 - z^2(\sigma)} \mathrm{d}\sigma + \varphi_0, \tag{12}$$

$$\psi = \int_0^s \frac{M_3 - M_z \cdot z(\sigma)}{1 - z^2(\sigma)} \mathrm{d}\sigma + \psi_0 + M_3 \cdot s \cdot \left(\frac{1}{b} - 1\right). \tag{13}$$

Since the system integral H is a function of θ and θ' [1] the set of quantities determining the configurations of the centerline of the tube is given by $\eta = \{F, M_z, M_3, \theta(0), \theta(0)'\}$.

Since the tube is clamped at $s = 0$ and can freely rotate at the other end ($s = L$) the boundary conditions imposed by the underlying problem are thus given as $\mathbf{r}(0) = \mathbf{x}_0$, $\mathbf{d}_1(0) = \mathbf{d}_{10}$, $\mathbf{d}_2(0) = \mathbf{d}_{20}$, $\mathbf{r}(L) = \mathbf{x}_L$ and $\langle \mathbf{d}_3(L), \mathbf{t}_L \rangle = 1$, where $\mathbf{x}_L$ and $\mathbf{t}_L$ are the coordinates of the point and the tangent to be matched at $s = L$.

3.2 Numerical computations

Thus the solution of this two point boundary value problem is reduced to the solution of the following set $\mathcal{F}(\eta) = 0$ of non-linear equations:

$$\begin{aligned} \mathbf{r}(L) - \mathbf{x}_L &= 0, \\ 1 - \langle \mathbf{d}_3, \mathbf{t}_L \rangle &= 0, \\ \theta_0 - \theta_0' &= 0. \end{aligned} \tag{14}$$

Standard numeric solution techniques [7,8] require that the Jacobian matrix is known numerically at every iteration point. These methods use numeric approximations to the partial derivatives at the iteration points, if those are not given as program code. As we have derived a symbolic expression for the function $\mathcal{F}$, we will show how we can come up with rather efficient code for the Jacobian, too.

4 Using the Jacobian Matrix in Symbolic Form

Using the common subexpression elimination algorithm of Maple the function $\mathcal{F}$ can be described by a computation sequence involving the following number of commands:

$$\begin{aligned} &43\,\mathrm{assignments} + 29\,\mathrm{additions} + 65\,\mathrm{multiplications} + 5\,\mathrm{divisions} \\ &\qquad + 19\,\mathrm{functions} + 5\,\mathrm{integrals} \end{aligned} \tag{15}$$

The 19 function evaluations consist of 9 trigonometric and square root functions and 10 instances of the Jacobi elliptic functions sn. The 5 integrals come from the necessity to have the centerline of the space curve available in a form that allows for boundary conditions at two distinct points, i.e. to explicitly carry the integration of the tangent vector [1,6].

Standard tools for automatic differentiation and also the automatic differentiation procedure available in Maple cannot handle integral operators in their inputs. Thus we could not use a straight-forward automatic differentiation approach on the computation sequence of $\mathcal{F}$ to obtain a computation sequence for the Jacobian matrix.

Whereas the symbolic differentiation algorithm of Maple can handle occurring integrals, the symbolic expression representing $\mathcal{F}$ was too large for a straight-forward symbolic differentiation.

However, with the following method we successfully derived symbolic computation sequences for the Jacobian matrix.

- We used auxiliary symbolic functions for the roots z_1, z_2, and z_3 of the cubic polynomial occurring on the right-hand-side of (10) and its partial derivatives.
- Using these auxiliary functions in the expression of $\mathcal{F}$ we could successfully compute the Jacobian matrix in symbolic form using Maple.
  This computation required several minutes of computation time.
- A computation sequence could be obtained by Maple generated from the expressions of the Jacobian and assignment of the expressions of the roots z_1, z_2, and z_3 and its partial derivatives to the auxiliary symbolic functions.
  Notice that all symbolic partial derivatives of the expressions of the roots z_1, z_2, and z_3 could be obtained by Maple easily.

Using the optimize function on computational sequences the result for computing the Jacobian required the following number of commands:

$$\begin{aligned} &260\,\text{assignments} + 174\,\text{additions} + 419\,\text{multiplications} \\ &\quad +31\,\text{divisions} + 76\,\text{functions} + 24\,\text{integrals} \end{aligned} \tag{16}$$

Notice that because of the symbolic differentiation rules for “special functions” used by Maple the computation sequence contains calls to various of Jacobi’s elliptic function and also to incomplete elliptic integrals of the second kind.

Remark. If equation (14) is solved via a corresponding minimization problem of a real valued function $m_{\mathcal{F}}$, then the gradient function of $m_{\mathcal{F}}$ can be expressed in symbolic form similarly. The computational costs for this gradient after common subexpression elimination is almost identical to the one for the Jacobian after common subexpression evaluation, i.e. is also given by the number of commands stated in (16).

References

1. Liu, S., Weber, A.: A symbolic-numeric method for solving boundary value problems of Kirchhoff rods. In Ganzha, V.G., Mayr, E.W., Vorozhtsov, E.V., eds.: Computer Algebra in Scientific Computing (CASC ’05). Volume 3718 of Lecture Notes in Computer Science., Kalamata, Greece, Springer-Verlag (2005) 387–398
2. Grégoire, M., Schömer, E.: Interactive simulation of one-dimensional flexible parts. In: SPM ’06: Proceedings of the 2006 ACM symposium on Solid and physical modeling, Cardiff, Wales, United Kingdom, ACM Press (2006) 95–103

3. Healey, T.J., Mehta, P.G.: Straightforward computation of spatial equilibria of geometrically exact Cosserat rods (2003) http://tam.cornell.edu/Healey.html.
4. Simo, J.C., Vu-Quoc, L.: On the dynamics in space of rods undergoing large motions – a geometrically exact approach. Computer Methods in Applied Mechanics and Engineering **66** (1988) 125–161
5. Henderson, M.E., Neukirch, S.: Classification of the spatial equilibria of the clamped elastica: numerical continuation of the solution set. International Journal of Bifurcation and Chaos **14** (2004) 1223–1239
6. Nizette, M., Goriely, A.: Towards a classification of Euler-Kirchhoff filaments. Journal of Mathematical Physics **40** (1999) 2830–2866
7. Hopkins, T.R., Phillips, C.: Numerical Methods in Practice – A Guide to the NAG Library. Addison-Wesley, Reading, MA, USA (1988)
8. Press, W.H., Teukolsky, S.A., Vetterling, W.T., Flannery, B.P.: Numerical Recipes in C++, Second Edition. Cambridge University Press (2002)