Large Exponents in Shooting Method for Thin Flexures

authors go here – Ben is first

[1]National Institute of Standards and Technology, 100 Bureau Drive, Gaithersburg, MD, USA

E-mail: `stephan.schlamminger@nist.gov`

Abstract.

In designing single-side clamped flexures as part of torsion balances, optical suspensions, or pendulums for scientific use structures become thin and semi-analytic calculations of their bending become infeasible with standard double precision. Semi-analytic calculations can be more efficent than finite element methods allowing faster design optimization. We provide simple analytical results which show that failure of float64 semi-analytic bending simulation is due to small angle exponential growth of the bending angle. The analytic solutions are used to provide timesaving guesses for applying the shooting method to bending with an arbitrary precision implementation of Runge-Kutta 45 integration. We resolve cases where standard double precision implementations fail.

Keywords: arbitrary precision, Euler-Bernoulli beam, Runge-Kutta, compliant mechanism

1. Introduction

The basic function of compliant mechanisms can be modeled by using shear-free beams. Several authors have sought semi-analytic models of these structures to decrease the need for computational resources [1–4]. We study the numerical solution of a single flexural element. Consider a flexure suspending a weight, $F_{w,0} = mg$, and additionally a deflecting force G_0 that acts at the end of the flexure. The fiber angle θ and moment M a distance s along the neutral axis depend on boundary conditions and uniform elastic modulus E. We assume the flexure is clamped at $s = 0$ which means its initial moment $M(0)$ and initial angle $\theta(0)$ are zero up to numerical error. The flexure has a length along its neutral axis L. The geometry of deformation can be determined by the relations

$$\frac{dM}{ds} = F_{w,0}\sin(\theta(s)) + G_0\cos(\theta(s)) \tag{1}$$

$$\frac{d\theta}{ds} = \frac{M(s)}{EI(s)} \tag{2}$$

on the moment and angle [4]. The geometry of the unbent flexure is captured by $I(s)$ the second moment of area of the cross-section perpendicular to the neutral axis. For the cases of a circular cross section and rectangular cross section there are formulae for moments,

$$I_\circ(s) = \frac{\pi}{4}r(s)^4 \tag{3}$$

$$I_\square(s) = \frac{bh(s)^3}{12}. \tag{4}$$

We show an axis perpendicular view of a varying cross section flexure we studied that is characterized by Equation 4 in Figure 1.

A numerical solution can be obtained by using standard double precision (64 bits, also referred to as float64) ordinary differential equation (ODE) solvers and one-sided boundary conditions. The boundary condition on the other end of the flexure, $\theta(L) = \theta_0$, is obtained by the shooting method. For example, A final bending angle of $\theta_0 = 1$ of a flexure is apparent at the curled end in Figure 2. Precision beyond float64 was used to solve for the deformation in Figure 2 because $M(0) \sim 10^{-596}$.

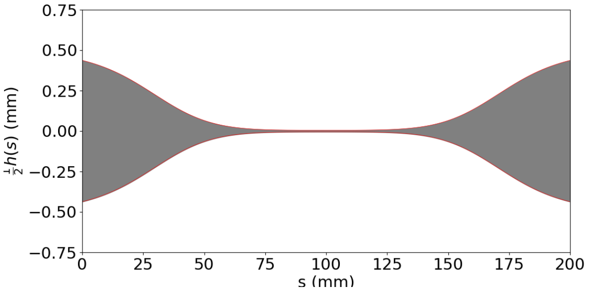


Figure 1: A thin planar flexure profile where for reasonable parameters ($E = 131 \times 10^9$Pa, $F_{w,0} = 1$N, $b = 0.1$mm) float64 is not sufficient when applying clamp-sided shooting. The flexure becomes 100 times smaller at its waist. Such a flexure should not fail to support the load $F_{w,0}$ if made of a high ultimate tensile strength alloy.

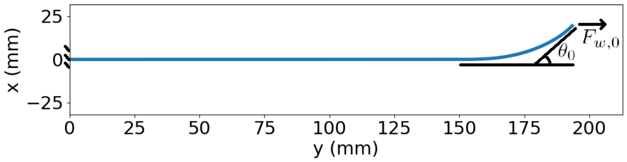


Figure 2: End torque bending to $\theta_0 = 1$ of a flexure with a rectangular cross section with varying $h(s)$. The parameters are those of Figure 1.

1.1. The shooting method

The shooting method is a technique in numerical ODEs where a double-sided boundary condition is met by guessing a free parameter. Shooting proceeds by iteratively solving the ODE trajectory given the free parameter and then improving the free parameter guess to better meet the second boundary condition. Iteration can be cast as root finding on an auxilliary function which encodes the desired boundary condition. In our case the root finding is on auxilliary function F,

$$F(M(0), G_0) = \theta(L, M(0), G_0) - \theta_0. \tag{5}$$

Evaluating the auxilliary function once requires numerically integrating the ODE once. Here our free parameter is either $M(0)$ or G_0. This approach is effective but when implemented with float64 it fails in very thin cases. Figure 1 shows a flexure geometry where a float64 solver would fail.

The purpose of this article is to numerically solve the bending of very thin flexures. For constant cross-sections, the system of differential equations can be solved analytically. In practice, however, the flexures are engineered to have a non-constant cross-section, and a numerical calculation is the only feasible technique to solve the bending of the flexure. The numerical solution of the bending equation can be solved many times faster than a finite element analysis. Hence, a fast solver can speed up the development cycle for compliant mechanisms. For flexures that are not super thin, these techniques have been used previously [1, 4].

1.2. Left versus right boundary condition as the free parameter

Typically [1, 4] the ODE integration step in the shooting method starts from the clamped end ($s = 0$). One iteratively targets the final bending angle θ_0. Alternatively, it should be feasible to shoot with parameter $M(L)$ from the end of the flexure with initial $\theta(L) = \theta_0$ and target $M(0) = 0$ without using accuracy above float64. For the sine-only case ($G_0 = 0$), where obvious semi-analytic approaches fail, one would need to achieve a reversed shooting condition. Namely $M(a) = 0$, where $\theta(x) = 0$ for all $x \leq a$. With this condition, we trivially guarantee $M(0) = 0$. For the nonzero G_0 case the condition is nearly as simple $M(a) = aG_0$. Using existing numerical libraries and integration with multicomponent simulations [4] could become more convoluted with this approach. As a result we choose to provide an algorithm for shooting from the clamped end.

1.3. Floating point representation

Limitations of numerical methods for differential equations based on fixed floating point are well known. For example Abad et al. show a unacceptably greater deviation from exact periodicity of numerical orbits with too few decimals of precision [5]. Double and single precision can be insufficient for numerically determining a choice of Lorenz attractor [6]. Limitations of floating point representation can also lead to the appearance of anomalous solutions which are only exhibited by the numerically approximated system [7].

A floating point number N is represented on a computer by two integers as $N = m \cdot 2^l$, named the mantissa and the exponent. As the name implies, float64 uses 64 bits to encode the binary N, the most significant bit is the sign bit. The sign bit is followed by an 11-bit wide exponent and a 52-bit wide mantissa. In double precision, the exponent lies between $-1022 \leq l \leq 1023$, limiting the calculations to those which have an order of magnitude between 10^{-308} and 10^{308}. For the calculation of bending the limitations of float64's exponent prevent calculations that appear in practical cases. As we shall discuss below, float64 is not sufficient for the calculations of very thin beams. Float128 extends this range enough for the application discussed here. Float128 uses a 15-bit wide exponent and a 112-bit wide mantissa. Together with the sign bit, the bit widths add up to 128 bits. Now, $-16382 \leq l \leq 16383$, which limits the exponents to 10^{-4932} and 10^{4932}. We use the library mpmath for Python to perform our calculations. It implements floating point calculations with an exponent range that exceeds the practical application of the bending model.

Table 1: Parameters of a circular cross section flexure used in an optical suspension [8].

Parameter	Value	Unit
E	7.3×10^{10}	Pa
L	6×10^{-1}	m
r	2×10^{-4}	m
$F_{w,0}$	1.47×10^{2}	N

2. Analytical solutions

2.1. Large exponents in bending

The problem that arises with exponent limitations in the outlined bending calculations is unintuitive because it involves scales smaller than the float64 minimum 10^{-308}. If something is so small it goes unmeasured. Numerically, this scale is of practical use because a symmetry-breaking nonzero initial moment $M(0)$ is needed when $G_0 = 0$. Without this small nonzero initial M the problem is symmetric and would not favor a leftward or rightward bend geometry. The solution shown in Figure 2 was determined by such a numerical trick. With unbroken symmetry ($\theta(0) = 0, M(0) = 0$), Equations 1 and 2 have null right-hand side. The result is zero bending for all s which carries over to numerical solutions.

To break symmetry we introduce negligible deviation from the physical boundary condition that the initial moment is zero. We must represent such small values in our ODE solver which is not possible with float64. For example, $M(0) \sim 10^{-331}$ to achieve a final bending angle of approximately unity radians with physical parameters for an optical suspension made of fused silica given in Table 1.

If $G_0 \neq 0$ then this numerical trick is not needed as the side force breaks the left-right symmetry of the system. According to the equation for the variation of θ (2) when the geometry becomes extremely thin ($I(s)$ is then small) the rate of change of θ will become extremely large. We will explicitly show the initial condition must be exponentially small in terms of the parameters of the geometry to achieve a physical end angle. This, in turn, explains why the above numerical trick employs extremely small numbers outside of float64's capability.

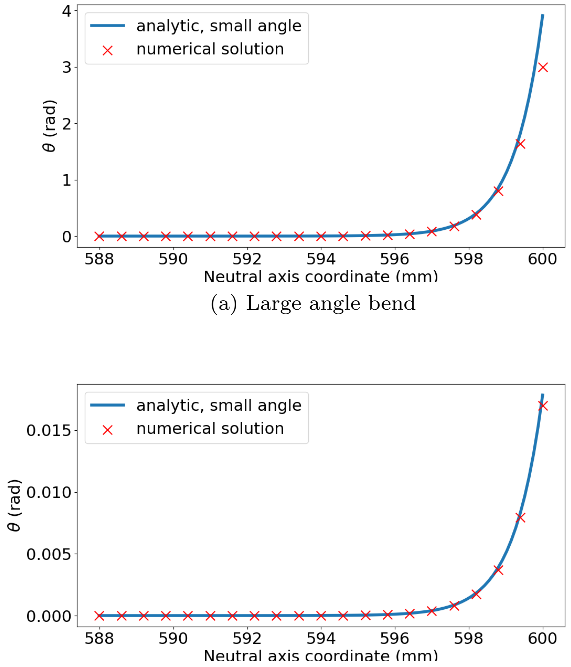
2.2. Small angle solutions

Sine term only To illustrate beyond a qualitative understanding that there is an exponential increase present in bending, consider the solutions where θ is small and $F_{w,0}\sin(\theta)$ dominates. Such hyperbolic trigonometric functions as solutions to beam deformation have been well documented in text books and are often used in practical research [9]. Here we need to look at these approximate solutions in a form that is amenable to providing a guess for shooting the bending ODE numerically. The angle is small for part of all bending geometries with $\theta(0) = 0$. Additionally, consider a constant width geometry $I(s) = I_0$. Then the equations for bending are simplified to

$$\frac{d^2\theta}{ds^2} = \frac{F_{w,0}}{EI_0}\theta. \tag{6}$$

The solution is trivially exponential. The scaling of θ from the initial $\theta(0)$ to $\theta(L)$ is easily quantified. We have

$$\theta(L) = \theta(0)\exp\left(\frac{L}{\lambda}\right) \text{ with } \lambda = \sqrt{\frac{EI_o}{F_{w,0}}}. \tag{7}$$



(a) Large angle bend

(b) Small angle bend

Figure 3: Bending angle near the end of a torque bending flexure with parameters as in Table 1. The small angle solution deviates from the numerical solver as the bend angle becomes sufficiently large.

We used this analytical solution to validate our numerical solver implementation. We choose the test case of a constant circular cross section, parameters in Table 1, under $G_0 = 0$ bending to final angles of 3 radians and 1°. When the angle of bending is still small the numerical result and the approximate analytic solution should coincide. Figure 3 shows this test case.

When making analytic guesses for shooting it is useful to estimate the total scaling over a varying geometry such as in Figure 1. We approximate the geometry as piecewise-constant. Such a piecewise constant approximation has been used in semi-analytic calculations of natural frequencies [3] of flexures and suffices as an initial guess in our application. The end angle will be

$$\theta(L) = \theta(0)\exp(\sqrt{\frac{F_{w,0}}{E}}\int_0^L \frac{1}{\sqrt{I(s)}}ds). \tag{8}$$

Calculating this scale factor for parameters of an optical suspension made of fused silica (provided in Table 1) the scale factor is order 10^{331}.

Cosine term only The same assumptions are made as in the sine case, only differing by considering a dominant $G_0\cos(\theta)$ term instead, and removing the piecewise-constant approximation. The approximate solution for small θ and $\theta(0) = 0$ is

$$\theta(L) = \int_0^L \frac{G_0 s}{EI(s)}ds. \tag{9}$$

General case We do not provide analysis for the general case. We empirically determined that applying the cosine approximation, and then scaling by the scale factor of the sine approximation provides a sufficient initial guess.

3. Shooting with a guessing algorithm

We assume the final bending angle is less than $\frac{\pi}{2}$ (extended to π for $G_0 = 0$) to avoid an oscillating regime on θ and M which could make shooting for solutions more difficult.

3.1. Guessing

The shooting method alone is sufficient to find bending solutions. We find in practice that it is much faster to provide an initial guess based on approximate solutions.

Sine term only guess The scale factor

$$\exp(\sqrt{\frac{F_{w,0}}{E}}\int_0^L \frac{1}{\sqrt{I(s)}}ds)$$

is calculated for the bending geometry. Then because there is error due to the piecewise approximation, the arbitrary precision Runge-Kutta 45(APRK45) solver is used to determine an ancillary bending angle $\theta^\star$ for an initial

$$M^\star(0) = \exp(-\sqrt{\frac{F_{w,0}}{E}}\int_0^L \frac{1}{\sqrt{I(s)}}ds).$$

Then the guess for the shooting stage is determined as $M(0) = \frac{\theta_0}{\theta^\star}M^\star(0)$. This is simply leveraging the linear nature of the approximate ODE to rescale the purely analytic estimate to better acheive θ_0 under the small angle approximation.

Cosine term only guess When a side force is present it determines the final bending angle and no numerical artifact needs to be introduced to break symmetry to cause a bend. Instead of shooting the initial condition $M(0)$, we have $M(0) = 0$ and are shooting G_0. In this case we can use the cosine-only approximate solution to figure that a good guess for G_0 to produce a bending through angle θ_0 is

$$G_0 = (\int_0^L \frac{s}{EI(s)}ds)^{-1}\theta_0.$$

Both terms significant In this case, G_0 breaks the symmetry, but the cosine-only solution does not provide a good guess. The math is hard, so we don't do it and instead take a guess which works well empirically. We find the scale factor for the sine-only solution, and multiply it on the initial guess for G_0 produced for a cosine-only estimate. That is

$$G_0 = (\int_0^L \frac{s}{EI(s)}ds)^{-1}\exp(-\sqrt{\frac{F_{w,0}}{E}}\int_0^L \frac{1}{\sqrt{I(s)}}ds)\theta_0.$$

Then run an iteration of APRK45 on this guess, and refine it by approximate linearity as in the sine term only estimate.

3.2. Shooting

Then a shooting step is done starting with the guess for either G_0 or $M(0)$ with the APRK45 solver for side force significant and side force insignificant cases respectively. We have found that the Anderson-Björck root finding method [10] was best applied to the shooting step, rapidly converging for a wide range of cases.

3.3. Speed and error

The fixed step-size APRK45 solver we implemented leverages fourth- and fifth-order coefficients to provide calculable and sufficiently small error. The geometry $I(s)$ is sampled at a fixed interval and cubic spline interpolated to feed the solver. Errors for our parameters were much smaller than our tolerance with APRK45 as shown in Figure 4 when run on a flexure with parameters as in Figure 1. These parameters can only be solved by shooting from the clamped end if we can represent $M(0) \sim 10^{-596}$. The time to do shooting for a single bend for the provided plots was about $5 - 50$ seconds on a personal computer with Python implementation.

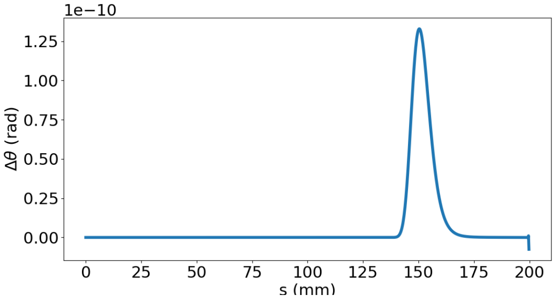


Figure 4: Point-wise error estimates for $\theta(s)$ solving $G_0 = 0$ bending with 1000 uniform Runge-Kutta steps for a ribbon as in Figure 1 with bending geometry shown in Figure 2.

4. Conclusions

For very thin flexures, we have determined that it is feasible to calculate their deformation using semi-analytic methods. This was shown using arbitrary precision floating point to exhibit limits of the double precision exponent bits. To integrate with existing tools in order to accommodate thin flexures double precision values can be exchanged for quadruple precision or arbitrary precision values. On the compiled language level this should not be difficult using ODE libraries which operate on generic types.

References

[1] L. Keck, S. Schlamminger, R. Theska, F. Seifert, and D. Haddad. Flexures for kibble balances: minimizing the effects of anelastic relaxation. *Metrologia*, 61(4):045006, Jul 2024.

[2] S. Henning and L. Zentner. Analytical characterization of spatial compliant mechanisms using beam theory. In *Microactuators, Microsensors and Micromechanisms*, pages 61–76. Springer International Publishing, 2023.

[3] V. Platl and L. Zentner. An analytical method for calculating the natural frequencies of spatial compliant mechanisms. *Mechanism and Machine Theory*, 175:104939, 2022.

[4] S. Henning and L. Zentner. Analysis of planar compliant mechanisms based on non-linear analytical modeling including shear and lateral contraction. *Mechanism and Machine Theory*, 164:104397, 2021.

[5] A. Abad, R. Barrio, and Á. Dena. Computing periodic orbits with arbitrary precision. *Phys. Rev. E*, 84:016701, Jul 2011.

[6] P. Wang, G. Huang, and Z. Wang. Analysis and application of multiple-precision computation and round-off error for nonlinear dynamical systems. *Advances in Atmospheric Sciences*, 23:758–766, 2006.

[7] E. Allen, J. Burns, D. Gilliam, J. Hill, and V. Shubov. The impact of finite precision arithmetic and sensitivity on the numerical solution of partial differential equations. *Mathematical and Computer Modelling*, 35(11):1165–1195, 2002.

[8] S. M. Aston, M. A. Barton, A. S. Bell, N. Beveridge, B. Bland, A. J. Brummitt, G. Cagnoli, C. A. Cantley, L. Carbone, A. V. Cumming, L. Cunningham, R. M. Cutler, R. J. S. Greenhalgh, G. D. Hammond, K. Haughian, T. M. Hayler, A. Heptonstall, J. Heefner, D. Hoyland, J. Hough, R. Jones, J. S. Kissel, R. Kumar, N. A. Lockerbie, D. Lodhia, I. W. Martin, P. G. Murray, J. O'Dell, M. V. Plissi, S. Reid, J. Romie, N. A. Robertson, S. Rowan, B. Shapiro, C. C. Speake, K. A. Strain, K. V. Tokmakov, C. Torrie, A. A. van Veggel, A. Vecchio, and I. Wilmut. Update on quadruple suspension design for Advanced LIGO. *Classical and Quantum Gravity*, 29(23):235004, December 2012.

[9] T. J. Quinn, C. C. Speake, and R. S. Davis. A 1 kg mass comparator using flexure-strip suspensions: Preliminary results. *Metrologia*, 23(2):87, jan 1986.

[10] N. Anderson and Å. Björck. A new high order method of regula falsi type for computing a root of an equation. *BIT*, 13:253–264, 1973.

[11] Z. Q. Wang, J. Jiang, B. T. Tang, and W. Zheng. High precision numerical analysis of nonlinear beam bending problems under large deflection. *Applied Mechanics and Materials*, 638:1705–1709, 2014.