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 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 efficient than finite element methods allowing faster design optimization. We provide simple analytical results which show that failure of double precision 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: double precision, 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 F_s 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 $s = 0$ with an initial tangent angle and bending moment. The flexure has a length along its neutral axis L. The geometry of deformation can be determined by the relations

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

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

on the moment and angle [4]. An example geometry, converted to cartesian coordinates, is shown in Figure 1. The geometry of the unbent flexure's cross section

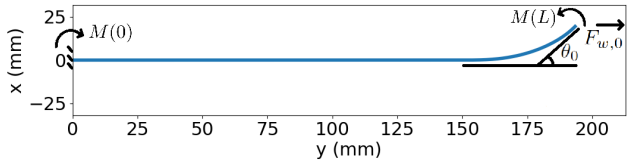


Figure 1: End torque bending to $\theta_0 = 1$ of a flexure with a rectangular cross section with varying $h(s)$. The parameters are those of Figure 2.

is captured by $I(s)$ the second moment of area of the plane perpendicular to the neutral axis. For the cases of a circular cross section and rectangular cross section there are formulae for second moments of area,

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

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

We show a neutral axis perpendicular view of a varying rectangular cross section flexure we studied that is characterized by Equation 4 in Figure 2.

A numerical solution can be obtained by using standard double precision (64 bits, also referred to as float64) ordinary differential equation (ODE) solvers that consider a single sided boundary condition. The boundary condition on the other end of the flexure, $\theta(L) = \theta_0$, is obtained by the shooting method. For

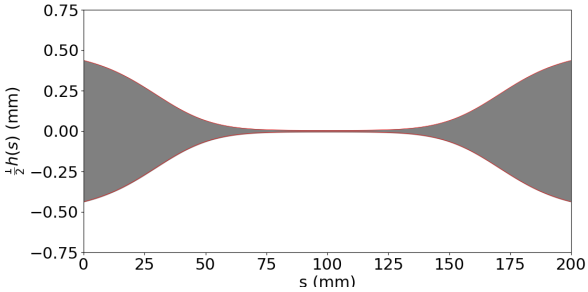


Figure 2: A thin planar flexure profile where for reasonable parameters ($E = 131 \times 10^9$Pa, $F_{w,0} = 1$N, $b = 0.1$mm) float64 is not sufficient when applying clamp-sided shooting. The flexure becomes 100 times smaller at its waist. Such a flexure should not fail to support the load $F_{w,0}$ if made of a high yield strength alloy such as precipitation hardened copper beryllium (> 1000 MPa).

example, A final bending angle of $\theta_0 = 1$ rad of a flexure is apparent at the curled end in Figure 1. The angle the fiber's tangent makes with the y axis at its end is θ_0 and thus is useful as a physically measurable boundary condition. Precision beyond float64 was used to solve for the deformation in Figure 1 because when numerically integrating the geometry $M(0) \sim 10^{-596}$ which is outside the capability of float64.

1.1. The shooting method

The shooting method reduces a boundary value problem to an initial value problem, where the end conditions can be iteratively satisfied by varying initial conditions [5]. An alternative approach to solving this problem would be to consider relaxation methods [5, 6]. 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 an auxilliary function F,

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

Evaluating the auxiliary function once requires numerically integrating the ODE. Here the parameter that is varied is either $M(0)$ or F_s, never both. This approach is effective but when implemented with float64 it fails in very thin cases. We define thin further as when the aspect ratio of the flexure profile becomes larger than a dimensionless quantity dependent on the Young's modulus and tensile strength. Figure 2 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 or constrained by specifications to have a non-constant cross-section, and a numerical calculation is the only feasible technique to solve the bending of the flexure. Nonlinear bending can be solved with finite element analysis or as an ODE, the latter much faster. Hence, an ODE-based 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]. Typically [1,4] the ODE integration step in the shooting method starts from the clamped end ($s = 0$) where boundary conditions are obvious.

1.2. Floating point representation

Limitations of numerical methods for differential equations based on fixed-size 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 [7]. Double and single precision can be insufficient for numerically determining a choice of Lorenz attractor [8]. Limitations of floating point representation can also lead to the appearance of anomalous solutions which are only exhibited by the numerically approximated system [9].

A floating point number N is represented on a computer by a natural number, an integer and a sign ($\pm$) as $N = \pm\frac{m}{2^{b'}} \cdot 2^{l}$. Multi-bit values m and l are called the mantissa and the exponent. The mantissa is normalized with b' the number of bits allocated to the mantissa. 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 [10]. 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. Quadruple precision or float128 extends this range enough for the application discussed here. Float128 uses a 15-bit wide exponent and a 112-bit wide mantissa [10]. 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}. One can represent slightly smaller values than listed by setting the mantissa to be small relative to $2^{b'}$, trading precision for magnitude. However full loss of mantissa precision spoils the algorithm we use to solve bending. 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 [11].

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 $F_s = 0$. Without this small nonzero initial M the numerical approximation problem is symmetric and would not favor a leftward or rightward bend geometry. The solution shown in Figure 1 was determined with a small initial value $M(0)$. 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 moment at the clamp. 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 order unity radians with physical parameters given in Table 1.

If $F_s \neq 0$ then this small moment 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 initial moments in zero side force bending are extremely small numbers outside of float64's capability.

2.2. Small angle solutions

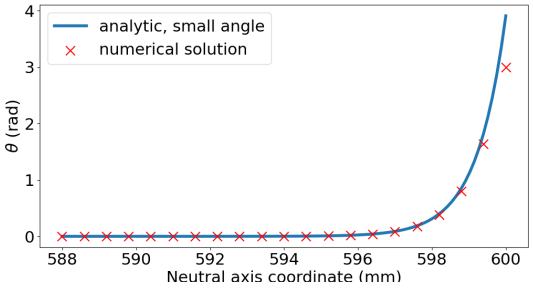
Sine term only To illustrate beyond a qualitative understanding that there is an exponential increase present in bending, consider the solutions where θ

is small and $F_{w,0}\sin(\theta)$ dominates. Such hyperbolic trigonometric functions as solutions to beam deformation have been well documented in textbooks and are often used in practical research [12, 13]. 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

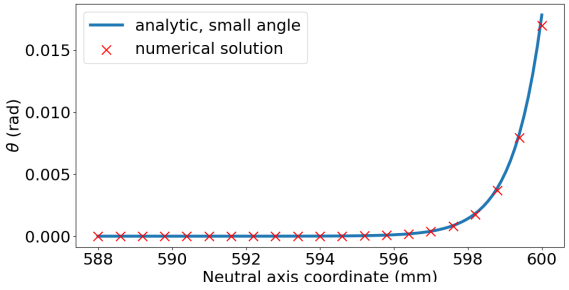
$$\frac{d^2\theta}{ds^2} = \frac{F_{w,0}}{EI_0}\theta. \tag{6}$$

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.

Definition of thin From this we can define what we mean by a thin flexure. Assume a circular cross section flexure is loaded with force $F_{w,0}$ which nearly exceeds the yield strength of the flexure σ_y. Further we have a large exponent in the scaling of the initial condition $\theta(0)$ or equally well $M(0)$. We have two conditions up to small proportionality constants,

$$\sigma_y = \frac{F_{w,0}}{r^2} \tag{8}$$

$$L\sqrt{\frac{F_{w,0}}{Er^4}} >> 1. \tag{9}$$

We then find a condition that may define a thin flexure as relevant to semi-analytic bending geometry for high sensitivity measurements:

$$\frac{L}{r} >> \sqrt{\frac{E}{\sigma_y}}. \tag{10}$$

If we bound and then ignore the material properties we arrive at the condition for a very slender beam. For a b bit floating point exponent we have a computational failure condition

$$\frac{L}{r} >> 2^b\sqrt{\frac{E}{\sigma_y}}. \tag{11}$$

Taking these calculations through without dropping small factors we provide Table 2 which informs one of the required floating point precision for solving bending.

We use the sine only analytical solution to validate our numerical solver implementation. We choose the test case of a constant circular cross section. Its parameters are given in Table 1. We calculate for $F_s = 0$ 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 initial guesses for shooting it is useful to estimate the total scaling over a varying geometry such as in Figure 2. 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

Table 2: Lookup table for bending solver IEEE754 precision. $L\sqrt{\frac{F}{min_s(EI)}} > \alpha$ means the precision is not sufficient. Minimizing EI over the fiber, $min_s(EI)$, provides a worse case estimate for deciding failure of the numerical precision.

IEE754	α
Single	87
Double	708
Quadruple	11 355
Octuple	181 702

suffices as an initial guess in our application. The end angle will be

$$\theta(L) = \theta(0)\exp(\sqrt{\frac{F_{w,0}}{E}}\int_0^L \frac{1}{\sqrt{I(s)}}ds). \tag{12}$$

Calculating this scale factor for parameters of an optical suspension made of fused silica (provided in Table 1) the scale factor is order 10^{330}.

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

$$\theta(L) = \int_0^L \frac{F_s s}{EI(s)}ds. \tag{13}$$

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 $F_s = 0$) 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 faster to provide an initial guess based on approximate solutions and linearity under small angle bending.

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 achieve θ_0 under the small angle approximation. This can be seen as a trivial case of shooting where the ODE is linear [5]. Approximate linearity is useful as it means our simple guesses only need to be good enough to underestimate the magnitude of the initial conditions. Then the rescaling using θ^* brings us much closer before applying full fledged shooting. In studying nonlinear bending we must do further shooting in all cases. We could also do two iterations of shooting, one to determine an order of magnitude for the initial condition and one to refine the mantissa. Here we use guesses by approximate linearity instead of shooting for the exponent.

Cosine term only guess When a side force is present it determines the final bending angle and a nonzero $M(0)$ need not be introduced to break symmetry causing a bend. Instead of shooting the initial condition $M(0)$ it is fixed and we are shooting F_s. We assume the cosine term dominates which is certainly true for small enough θ. In this case we can use the cosine-only approximate solution to figure that a good guess for F_s to produce a bending through angle θ_0 is

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

Both terms significant 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 F_s produced for a cosine-only estimate. That is

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

Then run an iteration of APRK45 on this guess, and refine it to a final guess 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 F_s 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 [14] 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

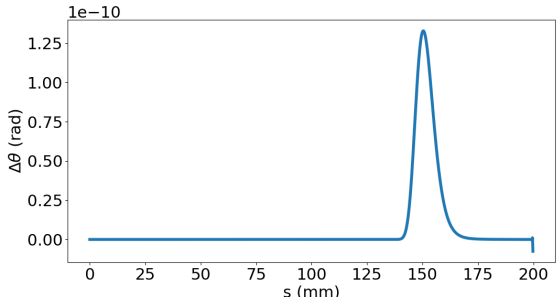


Figure 4: Point-wise error estimates for $\theta(s)$ solving $F_s = 0$ bending with 1000 uniform Runge-Kutta steps for a ribbon as in Figure 2 with bending geometry shown in Figure 1.

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 2. 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. Our implementation has been shared on github: usnistgov/BeamBending.

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. To integrate with existing tools in order to accommodate thin flexures double precision values can be exchanged for higher precision values according to Table 2. 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. *Mech. Mach. 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. *Mech. Mach. Theory*, 164:104397, 2021.

[5] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical Recipes: The Art of Scientific Computing*. Cambridge University Press, 3rd edition, 2007.

[6] N. Perrone and R. Kao. A general nonlinear relaxation iteration technique for solving nonlinear problems in mechanics. *J. Appl. Math. Mech.*, 38(2):371–376, Jun 1971.

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

[8] P. Wang, G. Huang, and Z. Wang. Analysis and application of multiple-precision computation and round-off error for nonlinear dynamical systems. *Adv. Atmos. Sci.*, 23:758–766, 2006.

[9] 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. *Math. Comput. Model.*, 35(11):1165–1195, 2002.

[10] IEEE Computer Society. IEEE Standard for Floating-Point Arithmetic. IEEE Std 754-2019, July 2019. DOI: 10.1109/IEEESTD.2019.8766229.

[11] 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. *Class. Quantum Gravity*, 29(23):235004, December 2012.

[12] 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.

[13] C. C. Speake. Anelasticity in flexure strips revisited. *Metrologia*, 55(1):114, Jan 2018.

[14] 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.