Stable numerical technique to calculate the bending of flexures with extreme aspect ratios

Benjamin Schreyer[1], Lorenz Keck[2], Jon R. Pratt[1], Stephan Schlamminger[1]

[1]National Institute of Standards and Technology, 100 Bureau Drive, Gaithersburg, MD, USA
[2] SLAC National Accelerator Lab, 2575 Sand Hill Rd, Menlo Park, CA 94025, USA

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

Abstract. Flexures in torsion balances and precision mechanisms often exhibit extreme aspect ratios, causing exponential scaling in Euler-Bernoulli bending models. Standard double-precision arithmetic cannot resolve the small initial conditions required for accurate solutions. This paper presents a semi-analytic method combining an efficient 1D bending model with adaptive Runge-Kutta-Fehlberg integration in arbitrary precision that overcomes this limitation. A quantitative criterion for when extended precision becomes necessary is established and an open-source Python implementation is provided, which remains stable even for flexures with extreme aspect ratios.

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

1. Introduction

Precision measurement systems often rely on compliant mechanisms to provide frictionless, repeatable motion. In applications ranging from the realization of the mass unit via Kibble balances [1–5] to gravitational wave detection [6] and micro-force metrology [7–10], flexures serve as the critical interface between a known force and a measurable displacement. The performance of these instruments is fundamentally limited by the accuracy with which the elastic behavior of the flexures can be modeled and subsequently measured.

In many high-performance metrological instruments, flexures are designed with extreme aspect ratios. These flexures have very long and thin geometries to minimize parasitic stiffness or to maximize sensitivity in a specific degree of freedom. However, this geometric optimization introduces a profound challenge in measurement science: a "computational gap" where the mathematical models used to predict instrument behavior become numerically unstable. This instability is not a failure of the underlying physics (Euler-Bernoulli beam theory) but a failure of the standard numerical representation (IEEE-754 double precision) to resolve the exponential scaling of the system's sensitivity.

From the perspective of the Guide to the Expression of Uncertainty in Measurement (GUM) [11], the reliability of a measurement result is intrinsically linked to the integrity of the mathematical model of the measurand. In instruments where a compliant mechanism translates an electrical or gravitational force into a measurable displacement, the flexure model constitutes a significant component of the "Type B" evaluation of uncertainty. For flexures with extreme aspect ratios, the numerical instability of standard double-precision solvers can introduce a computational bias that is easily mistaken for physical non-linearity or experimental noise.

The motivation for this work arises from the need to ensure that the contribution of numerical error to the total combined uncertainty remains negligible. Establishing a quantitative threshold for numerical failure is therefore essential for preserving the traceability of the measurement chain in high-sensitivity applications. High-sensitivity applications include the realization of mass standards via Kibble balances or the calibration of small-force standards. Without such a framework, the accuracy of the instrument becomes limited not by physical constraints, but by the digital representation of the model itself.

Accurate calculation of flexure deformation is a prerequisite for determining the sensitivity and uncertainty budget of precision balances. The well-posedness and stability of Euler-Bernoulli systems have been studied in depth [12, 13].

In order to calculate deformation of precision flexures, semi-analytic numerical methods that strike a balance between analytical solutions and full finite element analysis are described. FEA discretizes the full three-dimensional geometry using a mesh. To distinguish the methods discussed, they are referred to as *semi-analytic*, as they reduce the problem to solving a coupled ordinary differential equation (ODE). These approaches offer significant computational advantages in mechanical simulation. The accompanying analysis this paper presents allows these advantages to be reliably applied to modeling of precision measurement flexures.

Standard ODE solvers rely on fixed-precision floating-point arithmetic, as a consequence these solvers can fail when modeling shear-free beam bending with an *extreme aspect ratio.* A flexure has an extreme aspect ratio when its geometry causes standard bending computations to suffer from underflow or numerical instability. A formal criterion for this is derived in Sec. 2.3. In such cases, the internal bending moment required to model the deflection can span many orders of magnitude, exceeding the dynamic range of the IEEE 754 double-precision format. To overcome this limitation, a Runge-Kutta-Fehlberg integrator with arbitrary-precision floating-point arithmetic is implemented. This enables accurate modeling of flexures that lie beyond the capability of ODE solvers provided by common scientific programming languages. Guidance on the level of numerical precision required for a given geometry is also provided.

1.1. *The Single Flexure*

The bending of a loaded single flexural element under torque and transverse force is one of the simplest systems to analyze. Yet even this simple case can still challenge numerical methods, particularly because of precision loss and instability at extreme aspect ratios. Consider a flexure, illustrated in Fig. 1, that supports a weight $F_\mathrm{w} = mg$ and is subjected to a transverse force F_d at its free end ($s = L$). The variable s measures position along the neutral axis from $s = 0$ at the clamped end to $s = L$ at the free end. The angle relative to the vertical is denoted $\theta(s)$, while the internal bending moment is $M(s)$. Without loss of generality, $\theta(0)$ is set to zero as illustrated in Fig. 1. The deformation of the flexure is governed by coupled differential equations:

$$\frac{\mathrm{d}M}{\mathrm{d}s} = F_\mathrm{w}(s)\sin\theta(s) + F_\mathrm{d}(s)\cos\theta(s) \qquad (1)$$

$$\frac{\mathrm{d}\theta}{\mathrm{d}s} = \frac{M(s)}{E(s)I(s)}, \qquad (2)$$

where $E(s)$ denotes the elastic modulus and $I(s)$ the second moment of area about the neutral axis. In the simplest case, $E(s), F_\mathrm{w}(s)$, and $F_\mathrm{d}(s)$ are constant.

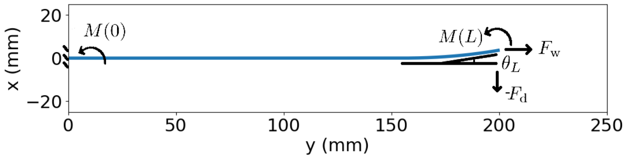


Figure 1: A loaded 200 mm long flexure is clamped at the top. The figure is shown rotated such that the flexure is lying on its side and gravity is acting on the $+y$ direction. A torque is applied such that the tangent at the end of the flexure is $\theta_L = 10°$. The coordinate s is measured along the neutral axis of the flexure and is distinct from the Cartesian axes.

The cross sectional geometry of the unbent flexure is characterized by $I(s)$, the second moment of area about the neutral axis. For a circular cross section with radius $r(s)$ and a rectangular cross section with width b and thickness $h(s)$, $I(s)$ is given by

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

$$I_\Box(s) = \frac{1}{12} b\, h^3(s), \qquad (4)$$

respectively. In general, as illustrated in Fig. 2, the cross sectional parameters vary with s, requiring a numerical solution for the resulting bending curve.

The flexure bending problem is solved as a two-point boundary-value problem using a shooting method [14,15]. An initial guess for $M(0)$ is iteratively updated so that the terminal condition $\theta(s = L) = \theta_L$ is satisfied within a user-specified tolerance. The value $M(L)$ follows directly from the solution, and F_w is treated as a constant specified by the user. Standard IEEE-754 double precision can only represent normal numbers down to $\approx 2.2 \times 10^{-308}$. The example shown in Fig. 1 requires resolving a bending moment ratio $M(0)/M(L) \sim 10^{-596}$, which lies well below this range and therefore underflows in double precision (float64). Consequently, higher-precision arithmetic beyond double precision (float64) is employed to obtain a stable solution.

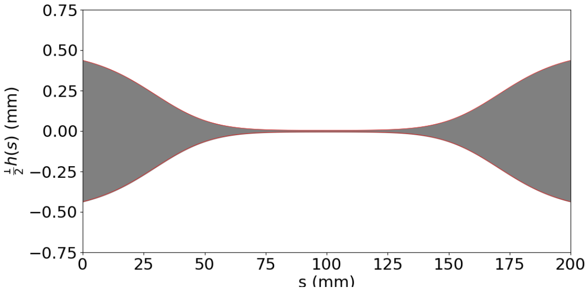


Figure 2: A thin planar flexure profile with parameters $E = 1.31 \times 10^5$ MPa, $F_\mathrm{w} = 1$ N, $b = 0.1$ mm. Standard double precision (float64) fails to maintain accuracy when applying clamp-sided shooting. The flexure thickness decreases by a factor of 100 at its waist.

A flexure with a constant cross section and no F_d provides a simple example of this numerical requirement, since its analytical solution is given by Speake (see Eq. (1) and Eq. (2) in [16]). It is

$$M(0) = 2M(L)e^{-\alpha L} \text{ with } \alpha^2 = \frac{F_\mathrm{w}}{EI}. \qquad (5)$$

The ratio $M(0)/M(L)$ becomes exponentially smaller as I decreases. This scaling soon exceeds the dynamic range of IEEE-754 double precision (float64), causing the shooting method to fail.

Although analytical solutions exist for flexures with constant cross sections, practical designs often involve non-uniform geometries, necessitating numerical approaches. Finite element analysis is a general method for such problems, but solving the underlying differential equations directly offers substantial computational advantages: the problem reduces to one dimension, avoiding full 3D meshing and enabling faster iteration during design. Among numerical strategies, ODE-relaxation methods [15, 17] and shooting techniques have both been applied successfully. The shooting method, in particular, has proven effective for flexures with moderate aspect ratios [4, 18], and this approach is extended to the extreme-aspect-ratio regime below.

1.2. The Shooting Method

To compute the neutral axis of a deflected flexure, the coupled bending equations, Eqs. (1–2), are numerically integrated. The shooting method is used to enforce boundary conditions, such as clamping and desired end angle $\hat{\theta}$. The shooting method transforms a boundary-value problem into a series of initial-value problems [14, 15], adjusting the initial condition, e.g., $p = M(0)$, until the boundary conditions are met. This iteration is formulated as a root-finding problem for the auxiliary function

$$G(p) = \theta_L(p) - \hat{\theta}, \qquad (6)$$

where $\theta_L(p)$ and $\hat{\theta}$ are the calculated and desired end angles, respectively. The shooting method, when implemented with double precision (float64), fails for flexures with extreme aspect ratios. In such cases, the required shooting parameter p underflows, preventing the root from being found. When this occurs, $\theta(L)$ diverges to nonphysical values (larger than π) for any nonzero initial condition, and the end-angle condition cannot be satisfied. Fig. 2 shows a geometry where this failure occurs. For simplicity, $F_\mathrm{d} = 0$, making $M(0)$ the sole shooting parameter that must match the desired boundary condition, $\theta_L(p) = \hat{\theta}$.

1.3. Floating-point Representation

Limitations of numerical methods that rely on fixed-size floating-point arithmetic are well known. Such issues arise in many contexts: for example, the restricted range of floating-point exponents has been documented in calculations involving Legendre polynomials [19, 20], and insufficient precision can cause numerical orbits to lose periodicity [21]. Double- or single-precision arithmetic may also fail to correctly resolve the behavior of chaotic systems, such as selecting among solutions of the Lorenz attractor [22]. More broadly, floating-point limitations can produce spurious solutions that are purely numerical artifacts [23]. In some cases, such errors can be detected and diagnosed automatically using dynamic program analysis techniques [24, 25].

A floating-point number N is represented in a computer using three components: a sign bit ($\pm$), an unsigned integer δ (bit width d) encoding the mantissa (or significand), and a signed integer λ (bit width l) encoding the exponent. For *normalized* numbers, the significand is interpreted as $1 + \delta/2^d$ so the value of the floating-point number is

$$N = \pm(1 + \frac{\delta}{2^d}) \cdot 2^\lambda. \qquad (7)$$

In the IEEE standard [26], double precision binary (float64) is encoded with 64 bits: 1 sign bit, $l = 11$ exponent bits, and $d = 52$ mantissa bits. The exponent is stored using a bias of $1023 = 2^{(l-1)} - 1$, resulting in a range $-2^{(l-1)} + 2 = -1022 \leq \lambda \leq 2^{(l-1)} - 1 = 1023$. spanning the representable values from $\approx 2.2 \times 10^{-308}$ to $\approx 1.8 \times 10^{308}$.

Subnormal numbers extend the range down to 5×10^{-324}. These occur when the exponent field is zero, which corresponds to $\lambda = -1022$. In that case, leading 1 is no longer implicit in the significand, and the value is computed as

$$N = \pm\frac{\delta}{2^d} \cdot 2^{-1022}. \qquad (8)$$

Subnormal numbers are a mechanism to extend the representable range below the smallest normalized value, but they come at the cost of reduced precision and are generally unsuitable for accurate numerical computation. Therefore, in this article, considered cases are limited to the dynamic range provided by normalized numbers.

The *dynamic range* of a floating-point number with l exponent bits is defined as the ratio of one to the smallest positive normal number. It is $2^{2^{l-1}}$. This definition of the dynamic range ignores the floating-point numbers with positive exponents, which is appropriate for the bending problem, since the end angle $\theta(L)$ is of order unity and hence $\theta(s)$ for $s < L$ utilizes the negative exponent range. With this definition, the problem can be solved without rescaling, see Section 1.4.

Quadruple precision (float128) uses $l = 15$ and, hence, provides a much larger dynamic range of $\approx 10^{4932}$, sufficient for the thin flexures considered here.

This larger exponent range of float128 enables the shooting method to resolve initial conditions that produce the desired end angles. The Python library mpmath [27] is therefore employed, which supports arbitrary-precision floating-point arithmetic, ensuring that the dynamic range is big enough for the physical problem, see Section 2.3.

1.4. Why Rescaling Cannot Replace Extended Precision

A common strategy to address numerical errors arising from extreme magnitudes is to rescale variables to a dimensionless form. For example, rather than computing the moment $M(s)$ in N m, one might work with the dimensionless ratio $M(s)/M_n$, where M_n is a reference moment, e.g., 1×10^{-20} N m. As shown in Eq. (5), the ratio $M(L)/M(0)$ between the free and fixed ends of the flexure can span hundreds of orders of magnitude. A single global scaling factor does not compress this exponential range, as it cancels out in relative calculations.

A possible solution would be to introduce a piecewise scaling function $\gamma(s)$ that adapts locally to the flexure geometry. However, this complicates the treatment of boundary conditions and may disrupt continuity. While such an approach could be developed, it is beyond the scope of this article.

Unlike moment rescaling, rescaling the angle is especially problematic due to the nonlinear behavior of trigonometric functions. In Taylor approximation, a $\theta(s)$ rescaled as $\theta(s) = \gamma\theta_n(s)$, yields

$$\cos\theta(s) = \cos\Big(\gamma\theta_n(s)\Big) \approx 1 - \frac{1}{2}\gamma^2\theta_n(s)^2. \qquad (9)$$

No single rescaling factor works across the entire flexure. Implementing piecewise scaling correctly is

Table 1: Parameters of a circular cross section flexure used in [6], assuming $F_\mathrm{d} = 0$. The parameters above the single line are the geometric and mechanical properties from which the parameters below the line are calculated.

Par.	Eq.	Value	
E		7.3×10^{10}	$\mathrm{N/m^2}$
L		6.00×10^{-1}	m
r		2.00×10^{-4}	m
F_w	mg	1.47×10^{2}	N
F_d		0	N
I	$\pi r^4/4$	1.26×10^{-15}	$\mathrm{m^4}$
α	$\sqrt{F_\mathrm{w}/(EI)}$	1.27×10^{3}	$\mathrm{m^{-1}}$
αL		7.60×10^{2}	
$\lfloor 2^{l-1} \ln 2 \rfloor_{50}$	for $l = 11$	7×10^{2}	
$e^{\alpha L}$		1.16×10^{330}	
σ_w	$F_\mathrm{w}/(r^2\pi)$	$1.176{\times}10^{9}$	$\mathrm{N/m^2}$
$\sqrt{E/\sigma_\mathrm{W}}$		7.90	
L/r		3.00×10^{3}	

nearly as complex as re-implementing trigonometric functions themselves.

In summary, rescaling alone cannot address the exponential dynamic range challenge, whereas increasing the floating-point exponent width offers a more robust and straightforward solution.

2. Analytical Solutions

Analytical solutions help explain why bending thin flexures becomes numerically difficult. They show how moments and angles can grow exponentially, making standard double-precision floating-point methods unreliable.

2.1. Large Exponents in Bending

The moment at the clamped end of the flexure is vanishingly small, yet a finite value is required to produce the correct shape from the coupled differential equations. If $F_\mathrm{d} = 0$ and $M(0) = 0$, the differential equations yield a straight flexure. To achieve bending of the flexure to either side, a small symmetry-breaking moment must be present. The solution in Fig. 1 was determined with a very small initial value $M(0)$.

For the flexure defined by the parameters in Table 1, a solution for $\theta(s)$ can be found by initiating the calculation with a very small moment $M(0)$. For example, achieving $\theta(L) = 1\,\mathrm{rad}$ requires $M(0) \approx 1 \times 10^{-330}\,\mathrm{N\,m}$, a value beyond the representable range of double precision (float64). The angles of the neutral fiber and bending moments exhibit exponential growth, which amplifies the limitations of floating-point precision.

2.2. Small-angle Solutions

Solutions for beam deformation involving hyperbolic trigonometric functions are well-documented in textbooks and are widely used in practical research [16,28]. Here, these solutions are revisited in a form that highlights their exponential nature. Any solution with $\theta(0) = 0$ will match the small-angle behavior over part of the flexure length. Therefore, the numerical limitations revealed by small-angle solutions also apply to large-angle bending problems.

In this example, a constant-width geometry is considered, with $I(s) = I_0$ and $\theta(0) = 0$. The case where $\alpha L \gg 1$ is applicable to characterize limits due to exponential behavior. Such an approximation is relevant for a highly loaded flexure based mechanism. For a cylindrical flexure loaded nearly to its yield strength Y, $\alpha = 2\sqrt{\frac{Y\pi r^2}{E\pi r^4}} = 2r^{-1}\sqrt{\frac{Y}{E}}$. Then, for steel, $\sqrt{\frac{Y}{E}} \sim 10$ and αL is about one order of magnitude greater than the flexure slenderness L/r.

The solutions of Speake [16] are adapted to the s, θ coordinate system, replacing his coordinate x with s, which is justified under the small-angle approximation. For $\alpha L \gg 1$, $\sinh(\alpha L) \approx \cosh(\alpha L) \approx \frac{1}{2}e^{\alpha L}$. Applying these to Eqs. (1) and (2) of [16] yields

$$M(L) \approx \frac{1}{2}\left(M(0) - \frac{F_\mathrm{d}}{\alpha}\right)e^{\alpha L}. \tag{10}$$

Using the compliance matrix of [16] gives

$$\theta(L) \approx \frac{F_\mathrm{d}}{F_\mathrm{w}}(1 - 2e^{-\alpha L}) + \frac{\alpha M(L)}{F_\mathrm{w}}. \tag{11}$$

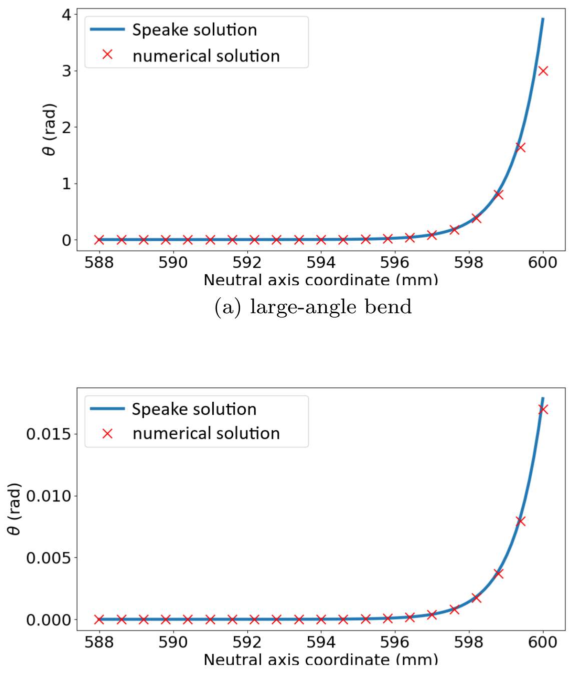
Substituting Eq. (10) into Eq. (11) gives

$$\theta(L) \approx \frac{F_\mathrm{d}}{F_\mathrm{w}}\left(1 - 2e^{-\alpha L} - \frac{e^{\alpha L}}{2}\right) + \frac{M(0)}{2F_\mathrm{w}}e^{\alpha L}. \tag{12}$$

Removing small terms from the parenthetical sum yields the approximation

$$\theta(L) \approx \frac{1}{2F_\mathrm{w}}\left(\alpha M(0) - F_\mathrm{d}\right)e^{\alpha L}. \tag{13}$$

This explicitly shows the exponential sensitivity of $\theta(L)$ to the tiny initial moment $M(0)$ and the small applied transverse load F_d, which drives the floating-point exponent limitations discussed above. Fig. 3 compares the exponential small-angle analytical solution for $\theta(s)$ with numerical calculations.



(a) large-angle bend

(b) small-angle bend

Figure 3: Bending of a loaded flexure under torque, using the parameters in Table 1, shown over the final 12 mm of the flexure. Panel (a) shows a large deflection, while panel (b) shows a small deflection. In panel (a), the breakdown of the small-angle approximation is evident as the small-angle solution deviates from the numerical result.

2.3. When Is a Flexure Considered Extreme Aspect Ratio?

The analytical solution above allows formal definition of the conditions under which a flexure is considered to have an extreme aspect ratio, relative to the available floating-point exponent range.

Consider a flexure with constant circular cross section loaded by a force F_w, resulting in a stress

$$\sigma_\mathrm{w} = \frac{F_\mathrm{w}}{\pi r^2}. \tag{14}$$

According to Eq. (13), the dynamic range required for the computation scales as $\exp(\alpha L)$ since parameters $F_\mathrm{d}, \alpha, M(0)$ must be represented numerically along with $\theta(L)$ which is exponentially larger. A flexure is defined as having an extreme aspect ratio if this required dynamic range exceeds the representable range of a binary floating-point number with exponent bit width l, i.e.,

$$e^{\alpha L} > 2^{2^{l-1}} \quad \longrightarrow \quad \alpha L > 2^{l-1}\ln 2. \tag{15}$$

Here, the $2^{2^{l-1}}$ reflects the maximum dynamic range permitted by the floating-point exponent, see Section 1.3.

For a circular cross section, this implies

$$\alpha L = L\sqrt{\frac{4F_\mathrm{w}}{E\pi r^4}} > 2^{l-1}\ln 2. \tag{16}$$

Combining Eq. (14) and Eq. (16) gives a compact criterion for when a flexure exceeds the dynamic range of floating-point formats:

$$\frac{L}{r} > \underbrace{\frac{\ln 2}{4}}_{0.17}\sqrt{\frac{E}{\sigma_\mathrm{w}}}2^l. \tag{17}$$

Now in pursuit of a condition for cases with varying cross section, a calculation remains stable and accurate if

$$\alpha L < 2^{l-1}\ln 2. \tag{18}$$

To provide extra margin for numerical stability, the right side of the inequality is rounded down to the nearest multiple of N_margin denoted by $\lfloor\cdots\rfloor_{N_\mathrm{margin}}$

$$\alpha L < \underbrace{\lfloor 2^{l-1}\ln 2\rfloor_{N_\mathrm{margin}}}_{\beta}. \tag{19}$$

For example with $N_\mathrm{margin} = 50$, $l = 11$, one obtains $2^{l-1}\ln 2 = 709.8$ and, hence, $\beta = \lfloor 2^{l-1}\ln 2\rfloor_{50} = 700$.

For a flexure with variable cross section, α becomes a function of arc length, and the inequality in Eq. (19) holding for α at its maximum value along the flexure $\alpha^\star$ is

$$\alpha^\star L = \max_{s\in[0,L]}\left(\sqrt{\frac{F_\mathrm{w}}{EI(s)}}\right)L \; < \; \beta. \tag{20}$$

Characterizing an embedded and more compliant beam with constant inverse length scale $\alpha^\star$ ensures that the numerically studied beam will not use more dynamic range than is available via the machine precision characterized by β. The more compliant embedded beam upper bounds the deflection and thus scaling of physical quantities associated with the numerically studied beam. More refined estimates for an embedded beam could be achieved with the WKB method but this is unnecessary given the limited number of choices of floating-point standards available.

Table 2 summarizes the floating-point precision required to satisfy this condition for various β (i.e., exponent bit–widths l). Example β values with $N_\mathrm{margin} = 50$ are provided, N_margin is small enough not to round any case to 0. For the bending parameters in Table 1, product $\alpha L = 760 > \beta_\mathrm{double} = 700$, so IEEE-754 double precision is inadequate for this case. To calculate this geometry, quadruple precision ($\beta_\mathrm{quadruple} = 11350$) is sufficient.

Table 2: IEEE 754 precision requirements for the bending solver. The condition that $\max_{s\in[0,L]}(\sqrt{\frac{F_\mathrm{w}}{EI(s)}})L < \beta$ ensures that the selected floating-point format provides sufficient dynamic range for accurate computation. The column labeled ϵ lists the smallest positive normalized number representable in each format.

name	total bits	exp. bits, l	ϵ	β $\lfloor 2^{l-1}\ln 2\rfloor_{50}$
single	32	8	1×10^{-38}	50
double	64	11	2×10^{-308}	700
quadruple	128	15	3.3×10^{-4932}	11 350
octuple	256	19	$1.5 \times 10^{-78\,913}$	181 700

2.4. Worst Case Sensitivity of a Nonlinear End Angle

When the bending angle is small, the system is linear and thus the the end bending angle is linearly related to the initial conditions. This is ideal sensitivity, since the precision and magnitude in the numerical representation of initial conditions directly maps into the end conditions. Near a large final bending angle, the variation of nonlinear trigonometric terms is reduced heavily from their linearization around their zero. Recall that linear approximate behavior about a trigonomitric zero is what causes the exponential growth which causes floating exponent limitations. For a worst case analysis, or lower bound on sensitivity, consider that variation in trigonometric terms on θ become small and can be approximated as constant which gives

$$\frac{dM}{ds} \sim F_\mathrm{w}, \tag{21}$$

$$\frac{d\theta}{ds} = \frac{M(s)}{EI(s)}. \tag{22}$$

It will become apparent that the nature of the end condition sensitivity derived from this approximation does not depend on assuming that $\sin\theta > 0$ or $\cos\theta > 0$ when they are approximated as constant. The choice to approximate $\frac{dM}{ds} \sim F_\mathrm{w}$ at saturation is for algebraic convenience. In any study with this approximation, the end angle grows exponentially until $s = L_\mathrm{NL}$ where the trigonometric nonlinearities saturate. Integrating Eq. (2) and Eq. (21), the remaining growth in the angle after saturation ($s > L_\mathrm{NL}$) assuming a constant cross section is

$$\theta(L) - \theta(L_\mathrm{NL}) \sim \frac{L - L_\mathrm{NL}}{EI}\left(M(L_\mathrm{NL}) + F_\mathrm{w}\frac{L - L_\mathrm{NL}}{2}\right). \tag{23}$$

Now write L_NL in terms of the initial conditions via assuming exponential growth to saturation at a large angle (unity)

$$\theta(L_\mathrm{NL}) \approx \frac{1}{2F_\mathrm{w}}(\alpha M(0) - F_\mathrm{d})e^{\alpha L_\mathrm{NL}} \approx 1 \tag{24}$$

$$-L_\mathrm{NL} \approx \ln\left(\frac{\alpha M(0) - F_\mathrm{d}}{2F_\mathrm{w}}\right)\frac{1}{\alpha}. \tag{25}$$

Finally substituting L_NL and $M(L_\mathrm{NL})$ via Eq. (10) into Eq. (23).

$$\begin{aligned}\theta(L) \sim 1 + \alpha L + \ln\left(\frac{\alpha M(0) - F_\mathrm{d}}{2F_\mathrm{w}}\right)& \\ + \frac{1}{2}\left(\alpha L + \ln\left(\frac{\alpha M(0) - F_\mathrm{d}}{2F_\mathrm{w}}\right)\right)^2&\end{aligned} \tag{26}$$

If we had chosen to approximate $\frac{dM}{ds} = 0$ instead of $\frac{dM}{ds} \sim F_\mathrm{w}$, only the square term would disappear. Therefore, the system has at worst logarithmic sensitivity when modulating the initial conditions to modify a nonlinear end angle. From this it may be understood that the initial condition exponent bits will play a key role in shooting for bending solutions to nonlinear angles. These bits are the rounded logarithm of the initial conditions. The conditioning of large-angle roots follows by differentiating, with respect to initial conditions. To apply this to a varying cross section one can take an upper bound on $I(s)$ of a highly compliant section of the flexure. Monotonicity dominating up to $\theta = \pi/2$ means cutting out parts of the flexure reduces sensitivity for acute bending.

3. Practical Tips for the Shooting Method

To compute a bending solution that satisfies a desired end angle $\hat{\theta}$, the shooting method is used, treating the initial bending moment $M(0)$ as the shooting parameter p. The goal is to determine the value of p such that the numerical solution yields $\theta(L) = \hat{\theta}$. A two-stage approach is followed, based on standard techniques from numerical analysis [14, 15].

The first stage begins by choosing $p^\star$, the smallest magnitude normal value in the floating-point format being used. Integrating the ODEs with this $p^\star$ yields a corresponding trial bending angle $\theta^\star = \theta_L(p^\star)$. Assuming the ODE system remains linear in this regime, the initial moment required to reach the target angle $\hat{\theta}$ can be estimated by scaling [14]:

If the desired angle is small, $\hat{\theta} < \theta_{\text{small}}$ with $\theta_{\text{small}} = 0.1\,\text{rad}$, the parameter is set as

$$p^\dagger = \frac{\hat{\theta}}{\theta^\star}\, p^\star. \tag{27}$$

Otherwise, to ensure convergence despite nonlinearity of trigonometric functions at large angles, θ_{small} is instead the target and the parameter set as

$$p^\dagger = \frac{\theta_{\text{small}}}{\theta^\star}\, p^\star. \tag{28}$$

This yields an initial condition that produces an end angle of the correct order of magnitude.

In the second stage, the linear estimate is refined using a nonlinear root-finding algorithm. Specifically, the Anderson-Björck method [29] is applied to solve $G(p) = \theta_L(p) - \hat{\theta} = 0$, with $p \in [p^\dagger/64, 64p^\dagger]$ around the linearized guess as the search domain.

This two-step process was successfully tested for robustness with a variety of flexure geometries. The first step results in a small, but correct search range for the second step, which then provides the correct, possibly tiny, $M(0)$. The algorithm works for arbitrary floating-point precision implemented by mpmath [27]. The pseudo-code for the algorithm is given below. The auxiliary function for root-finding during shooting, $G(p)$, may be evaluated for its worst case, large-angle, conditioning according to the sensitivity analysis of Sec. 2.4. Given the approximate behavior, Eq. (26), for any shooting parameter p which $M(0)$ and F_d depend on linearly

$$|G'(p)| > \left|\frac{1}{p}\right|. \tag{29}$$

Algorithm 1 Two step algorithm to find an initial condition p, e.g. $M(0)$, that gives a end angle $\hat{\theta}$. Subroutine APRKF45(p) numerically integrates the bending differential equation for initial condition p in arbitrary precision and returns the end angle $\theta_L(p)$.

```
procedure BEND TO $\hat{\theta}$, VARY $p$
    $\theta_{\text{small}} \leftarrow 0.1 rad$
    $p^\star \leftarrow smallestmagnitudefloating - pointvalue$
    $\theta^\star \leftarrow APRKF45(p^\star)$
    if $\hat{\theta} < \theta_{\text{small}}$ then
        $p^\dagger \leftarrow \frac{\hat{\theta}}{\theta^\star} p^\star$
    else
        $p^\dagger \leftarrow \frac{\theta_{\text{small}}}{\theta^\star} p^\star$
    $S \leftarrow [\frac{1}{64} p^\dagger, 64 p^\dagger]$
    return $root(APRKF45(p) - \hat{\theta})$ for $p \in S$
```

3.1. Computational Performance and Numerical Error

The arbitrary precision Runge-Kutta-Fehlberg 45 method (APRKF45) implemented leverages fourth- and fifth-order coefficients to provide calculable and sufficiently small errors. The varying cross sectional moment of inertia $I(s)$ is sampled at fixed intervals and interpolated using cubic splines for use by the solver. An arbitrary precision Runge-Kutta-Fehlberg 89 solver was also implemented, with eighth- and ninth-order coefficients [30, 31], which speeds up bending calculations for many flexure cross sections. Stepwise error estimates are provided by embedded methods. Errors for the chosen flexure parameters were much smaller than a practical angular tolerance with APRKF45. The stepwise error is shown in Fig. 4 for an analytically intractable bending problem. In Fig. 4 the estimated error is associated with solving $F_\text{d} = 0$ bending with 1000 uniformly spaced integrator steps for a ribbon as in Fig. 2 with bending geometry shown in Fig. 1. These parameters can only be solved by shooting from the clamped end if the computer system can represent $M(0) \sim 10^{-596}\text{N m}$. Numerical stability was verified by decreasing the step size and confirming convergence of the bending geometry. Test bending problems were solved in tens of seconds using our Python implementation. This runtime could be reduced to milliseconds by enabling Runge-Kutta-Fehlberg adaptive step-sizing and re-implementing the solver in a compiled language such as C++ or Fortran. The Python implementation used in this work is available at: https://github.com/usnistgov/BeamBending.

To validate the numerical solver, it is compared against the analytical solution for a constant cross section with $F_\text{d} = 0$, where the solution involves only the hyperbolic sine function. As a test case, a flexure with a constant circular cross section and the parameters listed in Table 1 is selected. This calculation requires a floating-point exponent range beyond double precision. Fig. 3 shows results for two target bending angles: 3 rad and 0.017 rad. The former represents an extreme deflection and clearly illustrates the breakdown of the small-angle approximation. In the latter case, the numerical and analytical solutions agree closely, as expected.

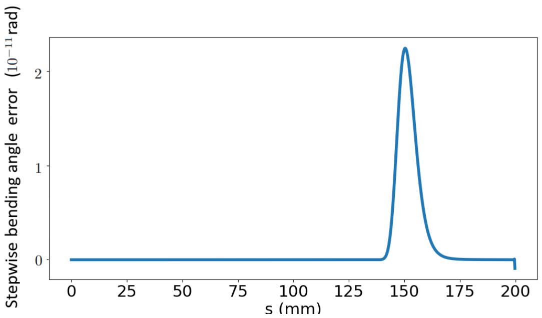


Figure 4: Local truncation error estimates at 1000 Runge-Kutta steps, $\Delta\theta$, for $\theta(s)$ generated by the embedded Runge-Kutta-Fehlberg 45 algorithm [30] when calculating bending for a flexure with nonuniform profile, see Fig. 2. If local truncation errors of a numerical method are similar in magnitude to the relevant variables, the solver has surely failed. This error estimate is different from a direct comparison with analytical solutions because one has no such analytical solution for the non-constant flexure cross section case.

4. Conclusions

This work presents a robust and efficient framework for simulating compliant mechanisms with extreme aspect ratio flexures. These are cases where standard double precision (float64) methods fail due to the exponentially large linear coefficient relating initial conditions and small-angle geometry intrinsic to Euler-Bernoulli bending. A clear numerical criterion determines when an extended exponent range is necessary.

Using arbitrary-precision Runge-Kutta-Fehlberg integrators, stable and accurate modeling is demonstrated in regimes where conventional solvers break down. Our open-source Python implementation supports fast and reliable flexure design across a wide range of geometries, making it a valuable tool for precision measurement and compliant mechanism applications.

Acknowledgments

We thank our NIST colleagues Jack Manley and John Lawall for their valuable feedback on the manuscript.

References

[1] I A Robinson and S Schlamminger. The watt balance: next generation. *Metrologia*, 53(5):A20, 2016.

[2] L. Keck. *Flexure-based mechanism for a Kibble Balance.* PhD thesis, Ilmenau, February 2025. Dissertation, Technische Universität Ilmenau, 2024.

[3] L. Keck, G. Shaw, R. Theska, and S. Schlamminger. Design of an electrostatic balance mechanism to measure optical power of 100 kW. *IEEE Transactions on Instrumentation and Measurement*, 70:1–9, 2021.

[4] 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, July 2024.

[5] O. R. Green, Y. Bao, J. R. Lawall, J. J. Gorman, and D. S. Barker. Accurate, precise pressure sensing with tethered optomechanics. *arXiv preprint*, arXiv:2409.00256:15, 2024.

[6] 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.

[7] Kumar Arumugam and Gordon Shaw. Perspective on small mass and force measurements. *Measurement Science and Technology*, 34(8):081002, may 2023.

[8] Kumar Arumugam, Jonathan Cripe, Sven Schulze, Stephan Schlamminger, and Gordon Shaw. Electrostatic force method to determine flexure stiffness with an integrated fiber-optic displacement interferometer. *Measurement Science and Technology*, 37(3):035001, jan 2026.

[9] Gordon A Shaw, Julian Stirling, John A Kramar, Alexander Moses, Patrick Abbott, Richard Steiner, Andrew Koffman, Jon R Pratt, and Zeina J Kubarych. Milligram mass metrology using an electrostatic force balance. *Metrologia*, 53(5):A86, sep 2016.

[10] J R Pratt, D T Smith, P R Nayfeh, and D B Newell. Progress toward a secondary traceable standard of force. *Journal of Research of the National Institute of Standards and Technology*, 110(6):667, 2005.

[11] Joint Committee for Guides in Metrology (JCGM). *Evaluation of measurement data – Guide to the expression of uncertainty in measurement (GUM).* BIPM, 1st edition, 2008. JCGM 100:2008.

[12] P. Deng, J. Zheng, and G. Zhu. Well-posedness and stability for a nonlinear Euler-Bernoulli beam equation. *Commun. Anal. Mech.*, 16(1):193–216, 2024.

[13] M. Sajjad Edalatzadeh and Kirsten A. Morris. Stability and well-posedness of a nonlinear railway track model. *IEEE Control Syst. Lett.*, 3(1):162–167, 2019.

[14] J. Stoer and R. Bulirsch. *Introduction to numerical analysis.* 3rd edition, 2002.

[15] 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.

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

[17] 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, June 1971.

[18] 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.

[19] J. M. Smith, Olver F. W. J., and D. W. Lozier. Extended-range arithmetic and normalized legendre polynomials.

ACM Trans. Math. Softw., 7:93–105, 1981.

[20] T. Fukushima. Numerical computation of spherical harmonics of arbitrary degree and order by extending exponent of floating point numbers. *J. Geod.*, 86(4):271–285, April 2012.

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

[22] 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.

[23] 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.

[24] F. Benz, A. Hildebrandt, and S. Hack. A dynamic program analysis to find floating-point accuracy problems. *SIGPLAN Not.*, 47(6):453–462, 2012.

[25] E. T. Barr, T. Vo, V. Le, and Z. Su. Automatic detection of floating-point exceptions. *SIGPLAN Not.*, 48(1), 2013.

[26] IEEE standard for floating-point arithmetic. *IEEE Std 754-2019 (Revision of IEEE 754-2008)*, pages 1–84, 2019.

[27] F. Johansson et al. mpmath: a Python library for arbitrary-precision floating-point arithmetic, 2023. Accessed: 2025-05-28.

[28] 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, January 1986.

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

[30] E. Fehlberg. New high-order runge-kutta formulas with step size control for systems of first-and second-order differential equations. *Zamm-zeitschrift Fur Angewandte Mathematik Und Mechanik*, 44, 1964.

[31] E. Fehlberg. Classical fifth-, sixth-, seventh-, and eighth-order runge-kutta formulas with stepsize control. Technical Report NASA TR R-287, National Aeronautics and Space Administration, Washington, D.C., 1968.

[32] 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.

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

[34] L. Keck, K. Arumugam, L. Chao, Z. Comden, F. Seifert, D.B. Newell, D. Haddad, and S. Schlamminger. Thoughts on the Kibble-Robinson theory. *Metrologia*, 62(2):025012, March 2025.

[35] T. L. Thomas, V. Kalpathy Venkiteswaran, G. K. Ananthasuresh, and S. Misra. Surgical applications of compliant mechanisms: A review. *Journal of Mechanisms and Robotics*, 13(2):020801, January 2021.

[36] Z. Wu and Q. Xu. Survey on recent designs of compliant micro-/nano-positioning stages. *Actuators*, 7(1), 2018.

[37] Z. Wang and H. Hu. Analysis and optimization of a compliant mechanism-based digital force/weight sensor. *IEEE Sensors Journal*, 5(6):1243–1250, 2005.

[38] Z. Cai, H. Wei, J. Chen, Y. Zhao, and Tian Y. Compliance and stress characteristics of the notch-type flexure hinges constructed on cylindrical beams. *Precision Engineering*, 95:515–525, 2025.