

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

Benjamin Schreyer¹, Lorenz Keck², Jon R. Pratt¹, Stephan Schlamminger¹

¹National 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 is established when extended precision becomes necessary. Furthermore, 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]. Semi-analytic methods bridge the gap between closed-form analytical solutions and computationally intensive finite element analysis (FEA). While FEA requires discretizing a full three-dimensional geometry, the semi-analytical approach reformulates the bending problem into a system of coupled ordinary differential equations (ODEs) along a single dimension. By collapsing the dimensionality of the task, these methods provide significant computational efficiency. The following analysis brings these computational gains to the design of precision flexures.

Standard ODE solvers use fixed-precision arithmetic and often fail when modeling shear-free beam bending with an extreme aspect ratio. In these cases, the internal bending moment 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. A formal criterion for these cases is derived in Sec. 2.3, along with guidance on the precision required for a given geometry.

1.1. The Single Flexure

A single flexure under torque and transverse force represents a fundamental system. However, even this basic configuration challenges numerical methods due to precision loss at extreme aspect ratios. Figure 1 illustrates a flexure supporting a load $F_w = mg$ and subjected to a transverse force F_d at its free end ($s = L$). The variable s measures distance along the neutral axis from the clamp $s = 0$ to the tip $s = L$. The angle $\theta(s)$ is measured relative to the vertical, with the clamped boundary condition $\theta(0) = 0$. The internal bending moment is denoted by $M(s)$. The deformation of the flexure is governed by coupled differential equations:

$$\frac{dM}{ds} = F_w \sin \theta(s) + F_d \cos \theta(s) \quad (1a)$$

$$\frac{d\theta}{ds} = \frac{M(s)}{E(s)I(s)}, \quad (1b)$$

where $E(s)$ denotes the elastic modulus and $I(s)$ the second moment of area about the neutral axis. For uniform beams, $E(s)$ and $I(s)$ are treated as constants.

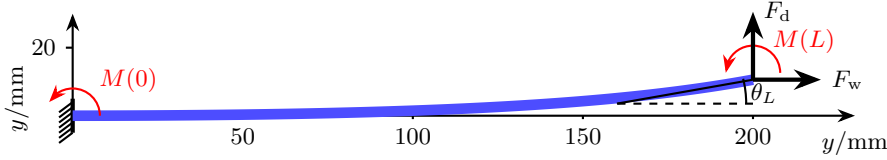


Figure 1. Schematic of a 200 mm flexure clamped at $s = 0$. In this coordinate system, gravity acts along the $+y$ direction. An applied tip moment produces a final angle of $\theta_L = 10^\circ$. The arc-length coordinate s follows the neutral axis.

The flexural stiffness is determined by the specific cross-sectional geometry. For circular sections of radius $r(s)$ and rectangular sections of fixed width b and varying thickness $h(s)$, the second moments of area are:

$$I_c(s) = \frac{\pi}{4} r^4(s) \quad \text{and} \quad (2a)$$

$$I_\square(s) = \frac{1}{12} b h^3(s). \quad (2b)$$

These parameters may vary with s , see Fig. 2, a dependency that precludes closed-form analytical solutions for the deflection profile.

The flexure bending is modeled as a two-point boundary-value problem and solved using a shooting method [14, 15]. The unknown initial moment $M(0)$ is iteratively adjusted until the terminal angle $\theta(s = L)$ matches the target $\hat{\theta}$. Standard IEEE-754 double precision can only represent normal numbers down to $\approx 2.2 \times 10^{-308}$. However, the geometry shown in Fig. 2 requires moments spanning $M(0)/M(L) \sim 10^{-596}$.

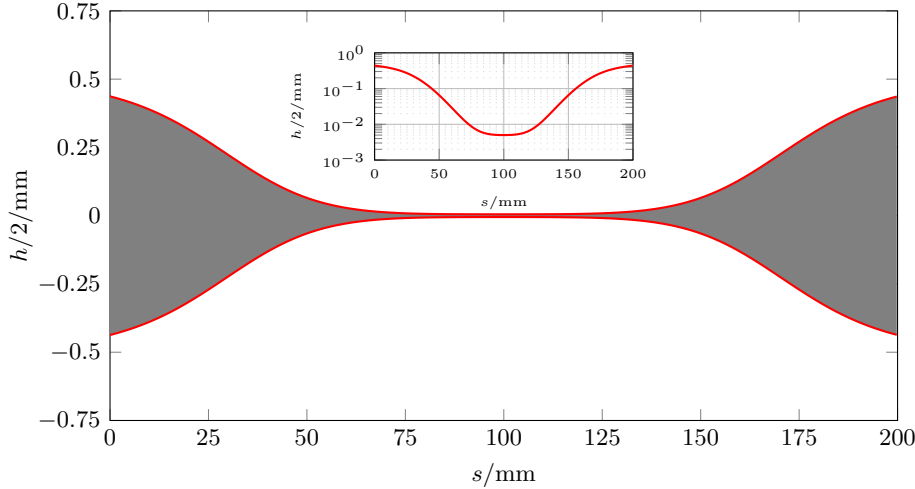


Figure 2. Profile of a planar flexure where the thickness reduces by a factor of 100 at the waist. The inset illustrates the positive half-thickness on a logarithmic scale to highlight this extreme geometric variation. Simulation parameters are $E = 1.31 \times 10^5$ MPa, $F_w = 1$ N, $b = 0.1$ mm. This extreme geometry causes standard double-precision (float64) shooting algorithms to fail.

This dynamic range causes standard double-precision (float64) solvers to underflow. Consequently, higher-precision arithmetic beyond double precision (float64) is required to obtain a convergent solution.

A uniform flexure with $F_d = 0$ illustrates the numerical challenge. The analytical solution given by Speake [16] is

$$M(0) = 2M(L)e^{-\alpha L} \text{ with } \alpha^2 = \frac{F_w}{EI}. \quad (3)$$

The ratio $M(0)/M(L)$ scales exponentially with decay length α^{-1} . For sufficiently small I or large L , this ratio exceeds the dynamic range of IEEE-754 double precision (float64), causing standard shooting methods to fail.

Analytical solutions are limited to uniform flexures, whereas practical designs often utilize non-uniform geometries. Numerical strategies for the resulting governing equations include relaxation [15,17] and shooting methods. While shooting techniques are effective for moderate aspect ratios [4,18], the following analysis extends the method to the extreme aspect ratio regime.

1.2. The Shooting Method

The coupled bending equations, Eq. (1), are numerically integrated to determine the deflection profile. A shooting method [14,15] enforces the boundary conditions, specifically the clamping angle and the target end angle $\hat{\theta}$. This technique transforms the boundary-value problem into an initial-value problem where the unknown initial moment $p = M(0)$ is iteratively adjusted. The iteration minimizes the residual

function:

$$G(p) = \theta_L(p) - \hat{\theta}, \quad (4)$$

where $\theta_L(p)$ is the integrated end angle for a given trial p . For flexures with extreme aspect ratios, standard double-precision (float64) arithmetic fails. The required initial moment p becomes so small that it underflows the floating-point representation. Consequently, the root-finder cannot locate the specific $M(0)$ required to satisfy the condition $\theta_L(p) = \hat{\theta}$. The flexure shown in Fig. 2 is a geometry where such failure occurs.

1.3. Floating-point Representation

The inherent limitations of fixed-precision arithmetic are well established. Restricted exponent ranges hinder calculations involving Legendre polynomials [19, 20], while insufficient precision causes numerical orbits to lose periodicity [21]. Standard precision also fails to capture the behavior of chaotic systems, such as the Lorenz attractor [22]. Precision issues may also lead to spurious solutions that are purely numerical artifacts [23]. While dynamic analysis can diagnose these errors [24, 25], the fundamental constraint remains the finite representation of the number system.

A floating-point number N comprises 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, noting an implicit leading bit for the significand:

$$N = \pm \left(1 + \frac{\delta}{2^d}\right) \cdot 2^\lambda. \quad (5)$$

The IEEE standard [26] allocates 64 bits for double precision binary (float64): 1 sign bit, $l = 11$ exponent bits, and $d = 52$ mantissa bits. The exponent employs a bias of $1023 = 2^{(l-1)} - 1$, yielding an effective normal range of $-2^{(l-1)} + 2 = -1022 \leq \lambda \leq 2^{(l-1)} - 1 = 1023$. This corresponds to a representable magnitude range from $\approx 2.2 \times 10^{-308}$ to $\approx 1.8 \times 10^{308}$.

Subnormal numbers extend the minimum representable magnitude to $\approx 5 \times 10^{-324}$. These values are encoded with a zero in the exponent field, corresponding to a fixed exponent $\lambda = -1022$ and no implicit leading bit. The value is given by

$$N = \pm \frac{\delta}{2^d} \cdot 2^{-1022}. \quad (6)$$

While this mechanism allows for gradual underflow, it sacrifices significant digits, rendering subnormals unsuitable for high-precision integration. Consequently, the following analysis is restricted to the dynamic range defined by normalized numbers.

The effective dynamic range of a format with l exponent bits is defined as the ratio of unity to the smallest positive normal number, approximately $2^{2^{l-1}}$. This definition excludes the positive exponent range, a restriction justified by the physics of the bending problem. Since the maximum deflection $\theta(L)$ is of order unity, the solution for $s < L$, $\theta(s)$ occupies the negative exponent range. Supporting this specific range allows the system to be solved directly without variable rescaling (see Sec. 1.4).

Quadruple precision (float128) with $l = 15$ offers a dynamic range of $\approx 10^{4932}$. This magnitude is sufficient for the extreme geometries analyzed in this work. To implement this capability, the Python library mpmath [27] is utilized. This library supports arbitrary-precision floating-point arithmetic, ensuring that the dynamic range encompasses the physical requirements detailed in Sec. 2.3.

1.4. Why Rescaling Cannot Replace Extended Precision

Variable rescaling is a standard technique for mitigating numerical errors. For instance, the moment $M(s)$ can be normalized by a reference moment M_n . However, as demonstrated in Eq. (3), the ratio $M(L)/M(0)$ spans hundreds of orders of magnitude. A global scaling factor shifts the absolute magnitudes but preserves this ratio. Therefore, if the ratio exceeds the intrinsic dynamic range of the floating-point format, global rescaling cannot prevent underflow or overflow.

Adaptive piecewise scaling $\gamma(s)$ could theoretically accommodate the range but introduces significant complexity regarding continuity and boundary conditions. Furthermore rescaling the angle is impeded by the nonlinearity of the trigonometric functions. For example, a Taylor approximation, of $\theta(s)$ rescaled as $\theta(s) = \gamma\theta_n(s)$, yields

$$\cos \theta(s) = \cos \left(\gamma\theta_n(s) \right) \approx 1 - \frac{1}{2}\gamma^2\theta_n(s)^2. \quad (7)$$

Rescaling cannot resolve the dynamic range deficit, leaving extended floating-point width the most practical solution.

2. Analytical Solutions

Analytical solutions illustrate the origin of the numerical instability encountered in thin flexures. These solutions reveal the exponential growth of moments and angles, rendering standard double-precision arithmetic insufficient.

2.1. Large Exponents in Bending

The moment at the clamp $M(0)$ is exceptionally small, yet it must remain finite to achieve the target deflection. When $F_d = 0$, a zero initial moment results in a trivial straight-beam solution. A non-zero moment is required to break the axial symmetry and initiate bending. The solution in Fig. 1 was determined with a very small initial value $M(0)$.

Using the parameters listed in Table 1, the deflection profile can be obtained by shooting from an exceptionally small initial moment $M(0)$. Specifically, achieving a target deflection of 1 rad requires $M(0) \approx 1 \times 10^{-330}$ N m. This magnitude lies beneath the representable range of double precision (float64). The exponential growth of the bending moment and fiber angle across the flexure length exacerbates the limitations of finite-precision arithmetic, rendering the solution unreachable without extended exponent range.

2.2. Small-angle Solutions

Solutions for beam deformation involving hyperbolic trigonometric functions are well established in the literature [16, 28]. Revisiting these solutions highlights the exponential growth that drives numerical instability. Because solutions satisfying the clamping condition $\theta(0) = 0$ track the small-angle limit near the fixed end, the numerical constraints identified in the linear regime persist in large-deflection problems.

A uniform flexure ($I(s) = I_0$) where $\alpha L \gg 1$ characterizes the exponential behavior.

Table 1. Parameters of a circular cross section flexure used in [6], assuming $F_d = 0$. Input properties are listed above the horizontal rule, with derived parameters shown below.

Par.	Eq.	Value
E		7.3×10^{10} 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} m ⁴
α	$\sqrt{F_w/(EI)}$	1.27×10^3 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	$F_w/(r^2\pi)$	1.176×10^9 N/m ²
$\sqrt{E/\sigma_w}$		7.90
L/r		3.00×10^3

The analytical solutions from Speake [16] are mapped to the (s, θ) domain, where the approximation $x \approx s$ is valid for small angles. In the regime $\alpha L \gg 1$, the hyperbolic functions simplify to $\sinh(\alpha L) \approx \cosh(\alpha L) \approx \frac{1}{2}e^{\alpha L}$. Applying these to Eqs. (1) and (2) of [16] yields

$$M(L) \approx \frac{1}{2} \left(M(0) - \frac{F_d}{\alpha} \right) e^{\alpha L}. \quad (8)$$

The terminal angle $\theta(L)$ is determined via the compliance matrix [16]

$$\theta(L) \approx \frac{F_d}{F_w} (1 - 2e^{-\alpha L}) + \frac{\alpha M(L)}{F_w}. \quad (9)$$

Substituting Eq. (8) into Eq. (9) relates the terminal angle directly to the initial moment,

$$\theta(L) \approx \frac{F_d}{F_w} \left(1 - 2e^{-\alpha L} - \frac{e^{\alpha L}}{2} \right) + \frac{M(0)}{2F_w} e^{\alpha L} \quad (10)$$

Removing small terms from the parenthetical sum yields the approximation:

$$\theta(L) \approx \frac{1}{2F_w} \left(\alpha M(0) - F_d \right) e^{\alpha L}. \quad (11)$$

This expression highlights the extreme sensitivity of $\theta(L)$ to the initial moment $M(0)$ and the transverse load F_d . In the regime of extreme aspect ratios, the $e^{\alpha L}$ term becomes so large that $M(0)$ must be infinitesimally small to yield a physical $\theta(L)$, precipitating the floating-point underflow described in Sec. 1.3. The agreement between this exponential approximation and the numerical results is illustrated in Fig. 3.

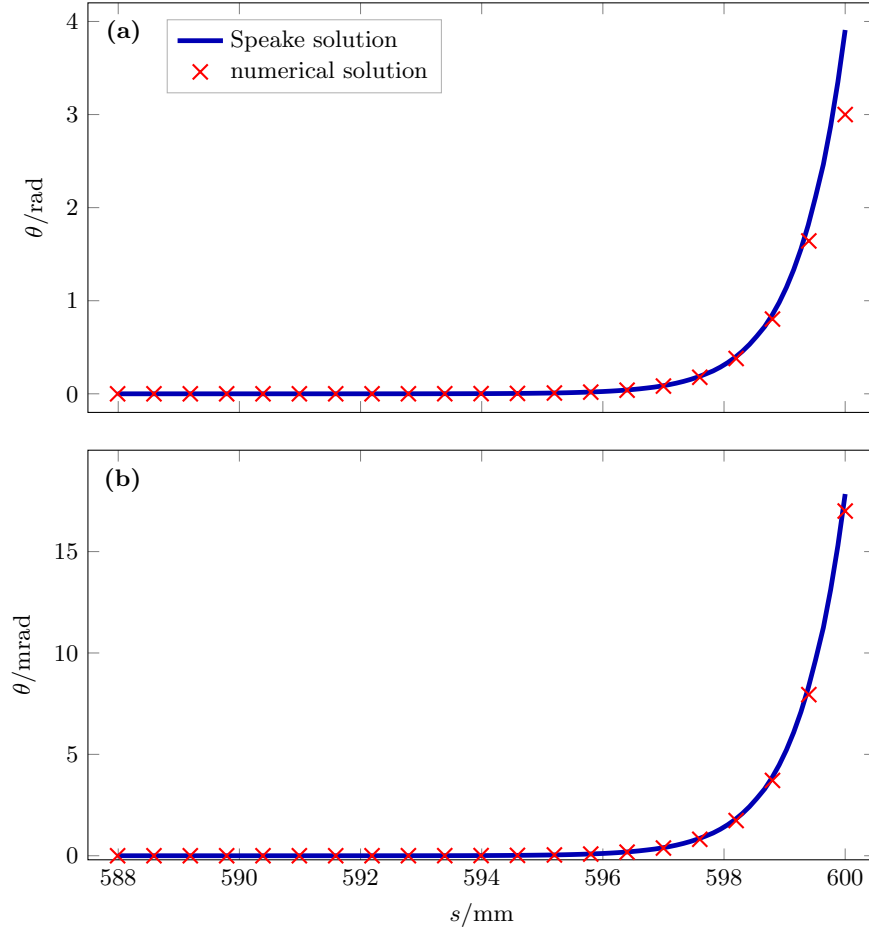


Figure 3. Bending of a loaded flexure under torque, using the parameters in Table 1, shown over the final 12 mm of the flexure. Panel (a) shows a large deflection, while panel (b) shows a small deflection. In panel (a), the breakdown of the small-angle approximation is evident as the small-angle solution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

$$\sigma_w = \frac{F_w}{\pi r^2}. \quad (12)$$

According to Eq. (11), the dynamic range required for the computation scales as $\exp(\alpha L)$ since parameters F_d , α , $M(0)$ 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.,

$$e^{\alpha L} > 2^{2^{l-1}} \quad \longrightarrow \quad \alpha L > 2^{l-1} \ln 2. \quad (13)$$

This threshold reflects the maximum dynamic range permitted by the floating-point exponent, see Sec. 1.3.

For a circular cross-section, the stability criterion implies:

$$\alpha L = L \sqrt{\frac{4F_w}{E\pi r^4}} > 2^{l-1} \ln 2 \quad (14)$$

Combining Eq. (12) and Eq. (14) yields a compact criterion for when a flexure exceeds the capacity of a given floating-point format:

$$\frac{L}{r} > \underbrace{\frac{\ln 2}{4}}_{0.17} \sqrt{\frac{E}{\sigma_w}} 2^l. \quad (15)$$

To generalize this condition for non-uniform geometries, the stability threshold is defined as the nearest multiple of a safety margin N_{margin} denoted by $\lfloor \dots \rfloor_{N_{\text{margin}}}$

$$\alpha L < \underbrace{\lfloor 2^{l-1} \ln 2 \rfloor_{N_{\text{margin}}}}_{\beta}. \quad (16)$$

For example with $N_{\text{margin}} = 50$, $l = 11$, 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, the condition must hold for the maximum value α^* :

$$\alpha^* L = \max_{s \in [0, L]} \left(\sqrt{\frac{F_w}{EI(s)}} \right) L < \beta. \quad (17)$$

This approach effectively compares the actual flexure to a hypothetical “worst-case” uniform beam with constant α^* . Since this hypothetical beam is more compliant than the actual geometry, it provides a conservative upper bound on the required numerical dynamic range. While more refined estimates could be derived using the WKB method, this simpler bound is sufficient given the coarse discretization of available floating-point standards.

Table 2 summarizes the floating-point precision required to satisfy the stability condition for various bit-widths l . The listed β values assume a safety margin $N_{\text{margin}} = 50$, ensuring the threshold remains non-zero for all standard formats. Applying this limit to flexure in Table 1, yields a product $\alpha L = 760$ which exceeds the double precision threshold $\beta_{\text{double}} = 700$. Consequently, IEEE-754 double precision is inadequate for this case, necessitating the use of quadruple precision ($\beta_{\text{quadruple}} = 11350$).

2.4. Numerical Conditioning of Large-Angle Bending Solutions

In the small-angle regime, the linear nature of the system ensures that the terminal angle $\theta(L)$ relates linearly to the initial conditions. This linear mapping provides

Table 2. IEEE 754 precision requirements for the bending solver. The condition that $\alpha^*L < \beta$ ensures that the selected 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

favorable numerical conditioning, as the precision of the initial parameters is preserved at the boundary. At large deflections, however, the trigonometric terms depart from their zero-centered linearizations and begin to saturate.

Recall that the exponential growth responsible for floating-point exponent limitations arises from the linear behavior near $\theta = 0$. To determine a lower bound on sensitivity in the nonlinear regime, we assume the trigonometric variations become small and can be approximated as constants. Here \sim indicates equality in sign and order of magnitude. This yields:

$$\frac{dM}{ds} \sim F_w, \quad (18a)$$

$$\frac{d\theta}{ds} = \frac{M(s)}{EI(s)}. \quad (18b)$$

For a uniform cross section, integrating these relations for the remaining growth after saturation ($s > L_s$, where the subscript s denotes the saturation point) gives:

$$\theta(L) - \theta(L_s) \sim \frac{L - L_s}{EI} \left(M(L_s) + \frac{F_w}{2}(L - L_s) \right) \quad (19)$$

At L_s , θ reaches unity and the exponential growth concludes:

$$\theta(L_s) \approx \frac{1}{2F_w} (\alpha M(0) - F_d) e^{\alpha L_s} \approx 1 \quad (20)$$

Solving for L_s yields

$$-L_s \approx \ln \left(\frac{\alpha M(0) - F_d}{2F_w} \right) \frac{1}{\alpha}. \quad (21)$$

Substituting L_s and $M(L_s)$ into Eq. (19) yields

$$\theta(L) \sim 1 + X + \frac{1}{2} X^2, \quad (22)$$

with

$$X = \alpha L + \ln \left(\frac{\alpha M(0) - F_d}{2F_w} \right). \quad (23)$$

This approximation demonstrates that the system exhibits, at worst, logarithmic sensitivity when modulating the initial conditions to achieve a nonlinear terminal

angle. Thus, the exponent bits of the floating-point representation play the decisive role in shooting for large-angle solutions, as these bits represent the logarithm of the initial condition. For non-uniform geometries, a conservative upper bound can be maintained by considering the most compliant sections of the flexure.

3. Practical Tips for the Shooting Method

To compute a bending solution for a desired end angle $\hat{\theta}$, a shooting method is employed, designating the initial bending moment $M(0)$ as the shooting parameter p . The objective is to find a value p for which the numerical solution satisfies $\theta(L) = \hat{\theta}$. A two-stage approach is implemented, following established numerical strategies [14, 15]. The first stage initializes the calculation with p^* , the smallest magnitude normal value in the floating-point format being used. Integrating the ODEs with this p^* yields a corresponding trial bending angle $\theta^* = \theta_L(p^*)$. Exploiting the linearity of the system at small angles, 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 $\theta_{\text{small}} = 0.1$ rad, the parameter is set as

$$p^\dagger = \frac{\hat{\theta}}{\theta^*} p^*. \quad (24)$$

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

$$p^\dagger = \frac{\theta_{\text{small}}}{\theta^*} p^*. \quad (25)$$

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 a variety of flexure geometries. The initial scaling phase effectively localizes the search range, ensuring that the second stage converges reliably to the required, often infinitesimal, value of $M(0)$. The algorithm is fully compatible with the arbitrary-precision arithmetic provided by `mpmath` [27], allowing for seamless scaling of the dynamic range as required by the flexure aspect ratio. The implementation logic is summarized in the pseudo-code below.

The conditioning of the auxiliary function $G(p)$ can be characterized for the large-angle regime following the sensitivity analysis in Section 2.4. Based on the approximation in Eq. (22), and assuming that $M(0)$ and F_d depend linearly on the shooting parameter p , the magnitude of the derivative satisfies:

$$|G'(p)| > \left| \frac{1}{p} \right|. \quad (26)$$

Hence, if there is an evaluation error ΔG in the objective function, this translates to a relative error in the computed parameter p that is strictly bounded by ΔG . This

Algorithm 1 Two step algorithm to find an initial condition p , e.g. $M(0)$, that gives an 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 $\theta_L(p)$.

```

1: procedure BEND TO  $\hat{\theta}$ , VARY  $p$ 
2:    $\theta_{\text{small}} \leftarrow 0.1\text{rad}$ 
3:    $p^* \leftarrow$  smallest magnitude floating-point value
4:    $\theta^* \leftarrow \text{APRKF45}(p^*)$ 
5:   if  $\hat{\theta} < \theta_{\text{small}}$  then
      $p^\dagger \leftarrow \frac{\hat{\theta}}{\theta^*} p^*$ 
6:   else
      $p^\dagger \leftarrow \frac{\theta_{\text{small}}}{\theta^*} p^*$ 
7:    $S \leftarrow [\frac{1}{64}p^\dagger, 64p^\dagger]$ 
8:   return root(APRKF45( $p$ ) -  $\hat{\theta}$ ) for  $p \in S$ 

```

demonstrates that the objective function is well-conditioned. Any numerical noise in evaluating the terminal angle does not severely inflate the error of the required initial moment.

3.1. Computational Performance and Numerical Error

The solver utilizes an arbitrary-precision Runge-Kutta-Fehlberg 45 (APRKF45) scheme, which employs fourth- and fifth-order coefficients to ensure bounded numerical errors. For non-uniform geometries, the varying moment of inertia $I(s)$ is sampled and reconstructed using cubic splines. To optimize performance, an APRKF89 variant utilizing eighth- and ninth-order coefficients [30, 31] was also implemented. This addition significantly accelerates calculations for many 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 $F_d = 0$ 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>.

The numerical solver is compared against the closed-form hyperbolic sine solution for a uniform beam with $F_d = 0$. A circular flexure with parameters listed in Table 1 serves as a test case. This specific geometry requires a floating-point exponent range beyond double precision. Figure 3 displays results for target angles of 3 rad and 0.017 rad. The larger deflection illustrates the expected breakdown of the small-angle

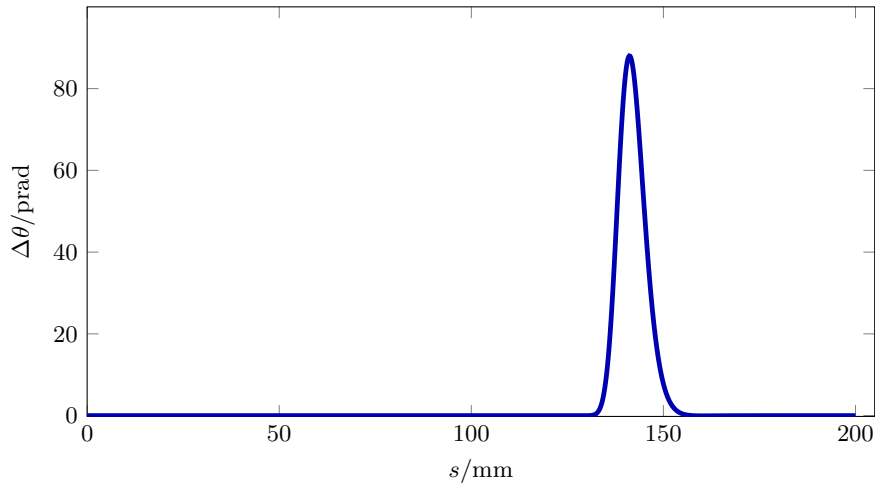


Figure 4. Estimated local truncation error $\Delta\theta$ across 1000 integrator steps for the profile $\theta(s)$, calculated using the embedded APRKF45 algorithm [30] for the non-uniform flexure geometry in Fig. 2. These internal error estimates provide a vital validation metric in the absence of a closed-form analytical solution for the variable cross-section case. As a baseline for reliability, the truncation errors remain several orders of magnitude below the target angular values.

approximation, while in the small-deflection regime, the numerical and analytical solutions are in excellent agreement.

4. Conclusions

This work establishes a robust and efficient numerical framework for simulating compliant mechanisms with extreme-aspect-ratio flexures. We identify the specific regimes where standard double precision (float64) is fundamentally insufficient due to the exponential sensitivity of the initial shooting parameter to the terminal boundary conditions. By formalizing the relationship between flexure geometry and numerical dynamic range, we provide a definitive criterion for the transition to extended-precision arithmetic.

The use of arbitrary-precision Runge-Kutta-Fehlberg integrators ensures stable and accurate modeling in regimes where conventional solvers fail. Our open-source Python implementation facilitates the fast, reliable design of non-uniform flexures across a wide range of geometries, providing a critical utility for the design of high-precision instruments and advanced compliant mechanisms.

Acknowledgments

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

References

- [1] Robinson I A and Schlamminger S 2016 *Metrologia* **53** A20
- [2] Keck L 2025 *Flexure-based mechanism for a Kibble Balance* Ph.D. thesis Technische Universität Ilmenau Ilmenau dissertation, Technische Universität Ilmenau, 2024 URL <https://doi.org/10.22032/dbt.63554>
- [3] Keck L, Shaw G, Theska R and Schlamminger S 2021 *IEEE Transactions on Instrumentation and Measurement* **70** 1–9
- [4] Keck L, Schlamminger S, Theska R, Seifert F and Haddad D 2024 *Metrologia* **61** 045006 URL <https://dx.doi.org/10.1088/1681-7575/ad57cb>
- [5] Green O, Bao Y, Lawall J, Gorman J and Barker D 2025 *Phys. Rev. Appl.* **24**(2) 024069 URL <https://link.aps.org/doi/10.1103/9dtb-sk2j>
- [6] Aston S M, Barton M A, Bell A S, Beveridge N, Bland B, Brummitt A J, Cagnoli G, Cantley C A, Carbone L, Cumming A V, Cunningham L, Cutler R M, Greenhalgh R J S, Hammond G D, Haughian K, Hayler T M, Heptonstall A, Heefner J, Hoyland D, Hough J, Jones R, Kissel J S, Kumar R, Lockerbie N A, Lodhia D, Martin I W, Murray P G, O'Dell J, Plissi M V, Reid S, Romie J, Robertson N A, Rowan S, Shapiro B, Speake C C, Strain K A, Tokmakov K V, Torrie C, van Veggel A A, Vecchio A and Wilmut I 2012 *Class. Quantum Gravity* **29** 235004 provided by the SAO/NASA Astrophysics Data System URL <https://ui.adsabs.harvard.edu/abs/2012CQGra...29w5004A>
- [7] Arumugam K and Shaw G 2023 *Measurement Science and Technology* **34** 081002 URL <https://doi.org/10.1088/1361-6501/acd134>
- [8] Arumugam K, Cripe J, Schulze S, Schlamminger S and Shaw G 2026 *Measurement Science and Technology* **37** 035001 URL <https://doi.org/10.1088/1361-6501/ae2afd>
- [9] Shaw G A, Stirling J, Kramar J A, Moses A, Abbott P, Steiner R, Koffman A, Pratt J R and Kubarych Z J 2016 *Metrologia* **53** A86 URL <https://doi.org/10.1088/0026-1394/53/5/A86>
- [10] Pratt J R, Smith D T, Nayfeh P R and Newell D B 2005 *Journal of Research of the National Institute of Standards and Technology* **110** 667
- [11] Joint Committee for Guides in Metrology (JCGM) 2008 *Evaluation of measurement data – Guide to the expression of uncertainty in measurement (GUM)* 1st ed (BIPM) jCGM 100:2008 URL <https://www.bipm.org/en/publications/guides/gum.html>
- [12] Deng P, Zheng J and Zhu G 2024 *Commun. Anal. Mech.* **16** 193–216
- [13] Edalatzadeh M S and Morris K A 2019 *IEEE Control Syst. Lett.* **3** 162–167
- [14] Stoer J and Bulirsch R 2002 *Introduction to numerical analysis* 3rd ed ISBN 978-1-4419-3006-4
- [15] Press W H, Teukolsky S A, Vetterling W T and Flannery B P 2007 *Numerical Recipes: The Art of Scientific Computing* 3rd ed (Cambridge University Press) ISBN 9780521880688
- [16] Speake C C 2018 *Metrologia* **55** 114 URL <https://dx.doi.org/10.1088/1681-7575/aaa112>
- [17] Perrone N and Kao R 1971 *J. Appl. Math. Mech.* **38** 371–376 ISSN 0021-8936 URL <https://doi.org/10.1115/1.3408785>
- [18] Henning S and Zentner L 2021 *Mech. Mach. Theory* **164** 104397 ISSN 0094-114X URL <https://www.sciencedirect.com/science/article/pii/S0094114X21001555>
- [19] Smith J M, Olver F W J and Lozier D W 1981 *ACM Trans. Math. Softw.* **7** 93–105 URL <https://api.semanticscholar.org/CorpusID:15787346>
- [20] Fukushima T 2012 *J. Geod.* **86** 271–285 ISSN 1432-1394 URL <https://doi.org/10.1007/s00190-011-0519-2>
- [21] Abad A, Barrio R and Dena A 2011 *Phys. Rev. E* **84** 016701 URL <https://link.aps.org/doi/10.1103/PhysRevE.84.016701>
- [22] Wang P, Huang G and Wang Z 2006 *Adv. Atmos. Sci.* **23** 758–766
- [23] Allen E, Burns J, Gilliam D, Hill J and Shubov V 2002 *Math. Comput. Model.* **35** 1165–1195 ISSN 0895-7177 URL <https://www.sciencedirect.com/science/article/pii/S089571770200078X>
- [24] Benz F, Hildebrandt A and Hack S 2012 *SIGPLAN Not.* **47** 453–462 ISSN 0362-1340
- [25] Barr E T, Vo T, Le V and Su Z 2013 *SIGPLAN Not.* **48** ISSN 0362-1340
- [26] 2019 *IEEE Std 754-2019 (Revision of IEEE 754-2008)* 1–84
- [27] Johansson F *et al.* 2023 mpmath: a Python library for arbitrary-precision floating-point arithmetic accessed: 2025-05-28 URL <https://mpmath.org/>
- [28] Quinn T J, Speake C C and Davis R S 1986 *Metrologia* **23** 87 URL <https://dx.doi.org/10.1088/0026-1394/23/2/002>
- [29] Anderson N and Björck A 1973 *BIT Numer. Math.* **13** 253–264
- [30] Fehlberg E 1964 *ZAMM - Journal of Applied Mathematics and Mechanics* **44** URL <https://api.semanticscholar.org/CorpusID:203026934>
- [31] Fehlberg E 1968 Classical fifth-, sixth-, seventh-, and eight-order runge-kutta formulas

with stepsize control Technical Report NASA TR R-287 National Aeronautics and Space Administration Washington, D.C. URL <https://ntrs.nasa.gov/api/citations/19680027281/downloads/19680027281.pdf>