SoftwareX 10 (2019) 100245

Contents lists available at ScienceDirect

SoftwareX

journal homepage: www.elsevier.com/locate/softx

Original software publication

Elfun18 – A collection of MATLAB functions for the computation of elliptic integrals and Jacobian elliptic functions of real arguments

Milan Batista

University of Ljubljana, Faculty of Maritime Studies and Transport, Portorož, Slovenia

ARTICLE INFO

Article history:
Received 30 November 2018
Received in revised form 2 May 2019
Accepted 2 May 2019

Keywords:
Special functions
Elliptic integrals
Jacobian elliptical functions
MATLAB

ABSTRACT

We outline a set of MATLAB functions that enable the computation of elliptic integrals and Jacobian elliptic functions for real arguments. The correctness, robustness, efficiency, and accuracy of the functions are discussed in detail. An example from elasticity theory is provided to illustrate the use of this collection.

© 2019 The Author. Published by Elsevier B.V. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).

Code metadata

Current code version	v1.3
Permanent link to code/repository used of this code version	https://github.com/ElsevierSoftwareX/SOFTX_2018_246
Legal Code License	GNU General Public **License**
Code versioning system used	*none*
Software code languages, tools, and services used	*MATLAB*
Compilation requirements, operating environments & dependencies	
If available Link to developer documentation/manual	https://www.researchgate.net/publication/323399074_elfun18_334_Elliptic_Integrals_Jacobian_Elliptic_Functions_and_Theta_Functions_-_Reference_Manual_v11
Support email for questions	milan.batista@fpp.uni-lj.si

Notation	Description	Relations
k	Elliptic modulus	
k'	Complementary modulus	$k^2 + k'^2 = 1$
m	Parameter	$m = k^2$
q	Elliptic nome	$q = \exp\left(-\pi K'/K\right)$
x		$x = \sin\phi = \operatorname{sn}(u\|m)$
α	Modular angle	$m = \sin^2\alpha$, $k = \sin\alpha$
β	Characteristic angle	$\nu = \sin\beta$
ν	Characteristic	
ϕ	Amplitude	$\phi = \operatorname{am}(u\|m)$

1. Introduction

Elfun18 is a collection of MATLAB functions that enable the computation of various elliptic integrals and Jacobian elliptic functions for real arguments. Altogether, the collection contains 70+ different functions (see the Appendix). In particular, the collection covers all elliptic integrals and Jacobian elliptic functions given in [1–5]. Most of these functions are included in the computer algebra programs Maple and Mathematica, and the computational environment MATLAB. In addition, several software libraries include elliptic integrals and elliptic functions (AMath/DAMath,[1] BOOST,[2] CERN,[3] elliptic,[4] DATAPLOT,[5] SLATEC,[6]

E-mail address: milan.batista@fpp.uni-lj.si.

1 http://www.wolfgang-ehrhardt.de/amath_functions.html.
2 http://www.boost.org/doc/libs/1_53_0/libs/math/doc/sf_and_dist/html/math_toolkit/special/.
3 https://root.cern.ch/doc/v608/group__SpecFunc.html.
4 https://www.rdocumentation.org/packages/elliptic/versions/1.3-7.
5 http://www.itl.nist.gov/div898/software/dataplot/elli_fun.htm.
6 http://sdphca.ucsd.edu/slatec_source/TOC.htm.

https://doi.org/10.1016/j.softx.2019.100245
2352-7110/© 2019 The Author. Published by Elsevier B.V. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).

Mathematics Source Library,[7] MPMATH,[8] and MATHCW[9]). As the motive for writing such a collection, each of these mentioned programs and libraries has its own limitations, either in the set of available functions or the choice of functional arguments. The reason for this is likely the maze of possibilities by which one can define elliptic integrals and functions. Thus, the elliptic integrals can be given in several different forms [6]. In addition, an argument of every elliptic integral or function is either the modular angle, the modulus, or the parameter [3,4]. It should also be noted that, for the Legendre form of elliptic integrals, some of these programs and libraries ignore the fact that these integrals are quasi-periodic functions and therefore provide incorrect results.

To clarify the above observations, we consider two examples. The elliptic integral of the first kind can be given in four different ways. In the Legendre form, we have the following [1]:

$$F(\phi, k) \equiv \int_0^\phi \frac{d\theta}{\sqrt{1-k^2\sin^2\theta}} \text{ or } F(\phi|m) \equiv \int_0^\phi \frac{d\theta}{\sqrt{1-m\sin^2\theta}} \tag{1}$$

and in the Jacobi form, we have

$$\mathbf{F}(x, k) \equiv \int_0^x \frac{dt}{\sqrt{\left(1-t^2\right)\left(1-k^2t^2\right)}} \text{ or}$$
$$\mathbf{F}(x|m) \equiv \int_0^x \frac{dt}{\sqrt{\left(1-t^2\right)\left(1-mt^2\right)}} \tag{2}$$

where k is the modulus and m is a parameter. Note that we use light italic F for the Legendre form and bold $\mathbf{F}$ for the Jacobi form. Researchers are typically sloppy when dealing with these elliptic integrals. For example, the formula $F\left(\frac{\theta}{2}, 2\right)$ can be given with the comment that $F(x, m)$ is an elliptic integral of the first kind. Because distinctions in the notations of integrals (1) and (2) are not standard, the interpretation of$F(x, m)$ may be a problem for casual users because, as examples, Maple implements $\mathbf{F}(x, k)$, Mathematica implements $\mathbf{F}(x|m)$, and MATLAB Symbolic Math Toolbox (abbreviated as MATLAB SMT) implements $F(\phi|m)$. For example, if $\theta = 1$, we have four possible results: $F\left(\frac{1}{2}, 2\right) = 0.6774\ldots$, $F\left(\frac{1}{2}|2\right) = 0.5514\ldots$, $\mathbf{F}\left(\frac{1}{2}, 2\right) = 0.8429\ldots$, and $\mathbf{F}\left(\frac{1}{2}|2\right) = 0.5841\ldots$

Certain problems may also cause the domain of the integrals. To obtain the real values of the integrals, most libraries restrict the range of m to $[0, 1]$ (for example, MATLAB's ellipke function). However, such a restriction is not necessary, and is not included in MATLAB SMT and Maple, for example. If $m > 1$, then $F(\phi|m)$ is real if $|\phi| < \sin^{-1}\frac{1}{\sqrt{m}}$. However, the programs mentioned above implement the integral for any real ϕ, and therefore return a complex value if the argument is outside the real domain. In certain cases, this may lead to unexpected results. Moreover, additional confusion creates an implementation that returns only the real part of the integral. A typical example is a complete elliptic integral of the first kind, which has complex values for $m > 1$; some libraries implement only the real value $\mathrm{Re}[\mathbf{K}(m)] = \frac{1}{\sqrt{m}}\mathbf{K}\left(\frac{1}{m}\right)$, which may create misleading results.

As the second example, we consider an elliptic integral of the second kind. Using the parameter m as the argument, we have three possible forms: Legendre, Jacobi, and Jacobi's second form [1,5].

$$E(\phi|m) \equiv \int_0^\phi \sqrt{1-m\sin^2\theta}\, d\theta, \mathbf{E}(x|m)$$

$$\equiv \int_0^x \sqrt{\frac{1-mt^2}{1-t^2}} dt, \varepsilon(u|m) \equiv \int_0^u \mathrm{dn}^2(t|m)\ dt. \tag{3}$$

From the Legendre form of an elliptic integral, we obtain Jacobi's second form through the substitution$\phi = \mathrm{am}(u|m)$. From Jacobi's form of integrals, we obtain the Legendre form of an elliptic integral using the substitution $x = \sin\phi$ and Jacobi's second form through the substitution$x = \mathrm{sn}(u|m)$. Here, sn is a Jacobi function and am is a Jacobi amplitude function. Of particular interest is quasi-periodic function $\varepsilon(u|m)$, which is found in many applications; however, to the best of our knowledge, neither of the mentioned packages and libraries contains such an implementation. Therefore, in the literature, authors usefully define $\varepsilon(u|m)$ as $E[\mathrm{am}(x|m)|m]$ or $\mathbf{E}[\mathrm{sn}(x|m)|m]$, which is mathematically correct. However, if we use these definitions for the calculation, we obtain an incorrect result outside the first period of $\varepsilon(u|m)$. Namely, it is well known that a function having a periodic function as an argument is a periodic function. This means that $E[\mathrm{am}(x|m)|m]$, for $m > 1$, and $\mathbf{E}[\mathrm{sn}(x|m)|m]$ are periodic functions and not quasi-periodic functions, as $\varepsilon(u|m)$ should be. This distinction can be seen from Fig. 1, where graphs of $\varepsilon(u|m)$ and $E[\mathrm{am}(x|m)|m]$ are shown. Note that we calculate $E[\mathrm{am}(x|m)|m]$ using Elfun18 as *mpEllipticE(mJacobiAM(x,m),m)* and MATLAB SMT as *ellipticE(jacobiAM(x,m),m)*. We can see from the figure that MATLAB SMT R2018a presumably incorrectly treats the quasi-periodicity of the function am and therefore produces an incorrect graph of $E[\mathrm{am}(x|m)|m]$.

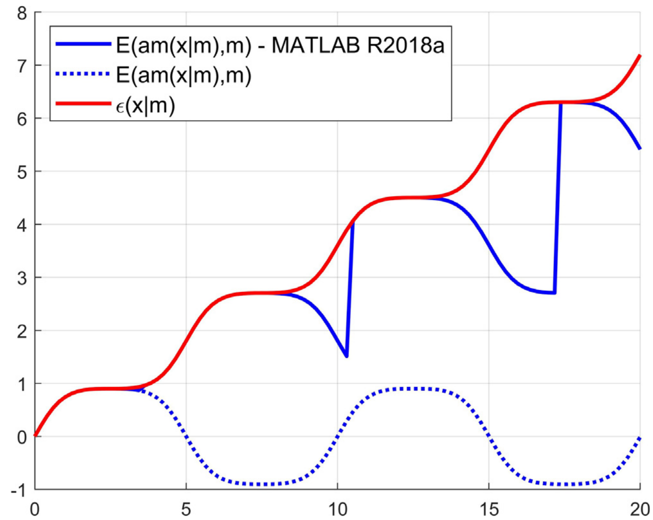

Fig. 1. Graph of $\varepsilon(x|m)$ and $E[\mathrm{am}(x|m)|m]$ when $m = 2$.

The primary purpose of Elfun18 is to provide a sufficiently comprehensive set of functions that allow the calculation of elliptic integrals and elliptic functions arising in practical problems, and that hopefully avoids the above-mentioned inconsistencies and errors. In the next section, we outline the functions. In Section 3, we describe the collection structure, naming conventions, and some calculation details. In Section 4, we discuss the collection performance, i.e., correctness, robustness, efficiency, and accuracy. Section 5 provides two examples from elasticity theory.

2. Functions

The collection includes the following classes of elliptic integrals and functions:

- Elliptic integrals

[7] http://www.mymathlib.com/functions/elliptic_integrals.html http://www.mymathlib.com/functions/elliptic_functions.html.

[8] http://docs.sympy.org/0.7.1/modules/mpmath/functions/elliptic.html.

[9] http://ftp.math.utah.edu/pub/mathcw/.

Table 1
Total execution time for computation of a complete elliptic integral of the first kind.

	10^6individual calls		Vector argument 10^4		Matrix argument 1000 × 1000	
	Time s	Ratio	Time s	Ratio	Time s	Ratio
melK/mEllipticK	0.524	1	1.270	1	1.273	1
elipke	3.235	6.17	0.230	0.18	0.210	0.17

Table 2
Total execution time for computation of Jacobian elliptic function *sn*.

	10^3 individual calls		Vector argument 1000		Matrix argument 100 × 100	
	Time s	ratio	Time s	Ratio	Time s	Ratio
mjsn/mJacobiSN	0.0025	1	0.0028	1	0.0226	1
ellipj	0.0286	11.3	0.0019	0.68	0.0102	0.45
jacobiSN	15.892	6269	0.896	324	7.981	352

- Bulirsch's elliptic integrals: $el1\left(x,k'\right)$, $el2\left(x,k',a,b\right)$, $el3\left(x,k',p\right)$, $cel\left(k',p,a,b\right)$
- Carlson's elliptic integrals: $R_C(x,y)$, $R_F(x,y,z)$ $R_G(x,y,z)$ $R_J(x,y,z,p)$ $R_D(x,y,z)$
- Incomplete elliptic integrals of the first, second, and third kind
    * Legendre's form: $B(\phi|m)$, $D(\phi|m)$, $F(\phi|m)$, $E(\phi|m)$, $\Pi(\phi,\nu|m)$
    * Jacobi's form: $\mathbf{B}(x|m)$, $\mathbf{D}(x|m)$, $\mathbf{F}(x|m)$, $\mathbf{E}(x|m)$, $\boldsymbol{\Pi}(\phi,\nu|m)$
    * Jacobi's second form: $\varepsilon(u|m)$, $\Lambda(u,\nu|m)$
- Complete elliptic integrals: $\mathbf{B}(m)$, $\mathbf{C}(m)$, $\mathbf{D}(m)$, $\mathbf{K}(m)$, $\mathbf{E}(m)$, $\boldsymbol{\Pi}(\nu|m)$
- Complimentary complete elliptic integrals: $\mathbf{K}'(m)$, $\mathbf{E}'(m)$, $\boldsymbol{\Pi}'(\nu|m)$
- Related functions:
    * Jacobi's zeta and omega functions: $Z(u|m)$, $\Omega(u,\nu|m)$
    * arithmetic-geometric mean: $\text{agm}(a,b)$
    * Heuman lambda function: $\Lambda_0(\beta|m)$

• Jacobian elliptic functions and their inverse

- principal Jacobian elliptic functions and their inverses: $\text{sn}(x|m)$, $\text{cn}(x|m)$, $\text{dn}(x|m)$
- subsidiary Jacobian elliptic functions and their inverses: $\text{cd}(x|m)$, $\text{sd}(x|m)$, $\text{nd}(x|m)$, $\text{dc}(x|m)$, $\text{nc}(x|m)$, $\text{sc}(x|m)$, $\text{ns}(x|m)$, $\text{ds}(x|m)$, $\text{cs}(x|m)$
- related functions:
    * amplitude function and its inversion: $\text{am}(x|m)$
    * Gudermannian function and its inverse: $\text{gd}(x)$
    * lemniscate functions and their inverses: $\text{sl}(x)$, $\text{cl}(x)$

• Theta functions

- Jacobi's theta functions: $\vartheta_1(x,q)$, $\vartheta_2(x,q)$, $\vartheta_3(x,q)$, $\vartheta_4(x,q)$
- Neville's theta functions: $\vartheta_s(x,q)$, $\vartheta_c(x,q)$, $\vartheta_d(x,q)$, $\vartheta_n(x,q)$
- Related functions:
    * elliptic nome and its inversion

The definitions and calling syntax for the above functions are given in the Appendix. More information on these functions can be found in [1,3,5]. Because we wrote a MATLAB collection, it is worth noting that MATLAB contains complete elliptic integrals K and E and principal Jacobian elliptic functions *sn*, *cn*, and *dn* for $0 \le m \le 1$ (elipke, ellipj). MATLAB SMT contains incomplete integrals $F(\phi|m)$, $E(\phi|m)$, and $\Pi(\phi,\nu|m)$; complete elliptic integrals $\mathbf{K}(m)$, $\mathbf{E}(m)$, $\boldsymbol{\Pi}(\nu|m)$, $\mathbf{K}'(m)$, $\mathbf{E}'(m)$, and $\boldsymbol{\Pi}'(\nu|m)$; all principal and subsidiary Jacobian elliptic functions; and functions $\text{am}(x|m)$, $Z(u|m)$, and elliptic nome.

Table 3
Average total time for the solution of $2E(k) = K(k)$ using MATLAB function *fzero*.

	Timems	Ratio
elK/elF	0.46	1
ellipke	0.52	1.12

Whereas Bulirsch's and Carlson's elliptic integrals and one argument function from the above list have standard definitions, the definitions of other functions deserve some comments. The incomplete elliptic integrals and Jacobi elliptic functions depend on two arguments, except for Π, which depends on two or three arguments: one space argument and one (or two) elliptic arguments. The elliptic argument is either the modular angle α, the modulus $k = \sin\alpha$, or the parameter $m = k^2$. Therefore, three different notations are used [3]:

$$Q(x\backslash\alpha),\ Q(x,k),\ Q(x|m)$$

where Q is an incomplete elliptic integral or Jacobi elliptic function.

In all computational functions in Elfun18, we use the parameter m. The reasons are the following. First, using the modular angle α reduces the values of k to $-1 \le k \le 1$ and the values of m to $0 \le m \le 1$. To obtain the values of an elliptic function for k and m outside these intervals, we must use reciprocal-modulus transformation formulas [1,3]. Second, elliptic integrals and Jacobian elliptic functions are symmetric with respect to a real modulus k, that is,

$$Q(x,-k) = Q(x,k) \tag{4}$$

However, if the modulus is a pure imaginary number, the symmetry does not hold. In this case, $(ik)^2 = -k^2 = m$, i.e., parameter m is real. Therefore, we have

$$Q(x,ik) = Q\left(x|-k^2\right) = Q(x|m) \tag{5}$$

Using the parameter m thus covers all real moduli and all pure imaginary moduli. Further, if we have a function $Q(x|m)$, we can define

$$Q(x,k) \equiv Q\left(x|k^2\right) \tag{6}$$

Table 4
Accuracy test with random input. Comparison with MATLAB SMT functions ellipticE, ellipticF, and ellipticPi. The number of runs is 1000. MEA is the maximal relative error, MRE is the maximal relative error, RMS is the root mean square error, and $eps \approx 2.22\times 10^{-16}$.

Func.	min(|x|)	max(|x|)	min(m)	max(m)	MAE/eps	MRE/eps	RMS/eps
mpEllipticE	4.3E−04	5.0E−01	−5.0E−01	5.0E−01	1.5	4	1.2
mpEllipticE	1.0E−04	1.6E+01	−1.6E+01	1.6E+01	158	51	2.1
mpEllipticE	1.5E−06	5.0E+02	−4.2E+02	4.2E+02	5.45E+04	688	37.5
mpEllipticE	1.9E−07	1.6E+04	−5.1E+03	5.3E+03	3.80E+06	973	61.3
mpEllipticF	8.2E−04	5.0E−01	−5.0E−01	5.0E−01	1	3	0.9
mpEllipticF	2.4E−05	1.6E+01	−1.6E+01	1.5E+01	48	3	1.0
mpEllipticF	5.9E−06	5.0E+02	−1.7E+02	3.7E+02	768	3.5	0.9
mpEllipticF	3.4E−08	1.6E+04	−1.0E+04	1.6E+04	4.10E+04	3	0.9
mpEllipticPi	5.1E−05	5.0E−01	−5.0E−01	5.0E−01	5	3	1.1
mpEllipticPi	1.2E−05	1.6E+01	−1.6E+01	9.9E−01	32	9	1.2
mpEllipticPi	4.9E−06	4.8E+02	−5.0E+02	1.0E+00	32	25	2.5
mpEllipticPi	3.4E−08	1.3E+04	−6.5E+03	9.9E−01	64	112	8.8

Table 5
Accuracy test, $x = 0.23$, $k = 0.999$, $N = 10^5$.

Function	f(x)	f(x+N*K(k))	Diff.
jcd	0.999946	0.999946	1.42E−12
jcn	0.974120	0.974120	6.56E−10
jcs	4.309650	4.309650	5.68E−08
jdc	1.000050	1.000050	−1.42E−12
jdn	0.974172	0.974172	6.54E−10
jds	4.309880	4.309880	5.68E−08
jnc	1.026570	1.026570	−6.91E−10
jnd	1.026510	1.026510	−6.89E−10
jns	4.424150	4.424150	5.53E−08
jsc	0.232037	0.232037	−3.06E−09
jsd	0.232025	0.232025	−3.06E−09
jsn	0.226032	0.226032	−2.83E−09
jzeta	0.174671	0.174671	1.53E−11

$$Q\left(x\backslash\alpha\right) \equiv Q\left(x|\sin^2\alpha\right) \qquad (7)$$

Thus, $Q(x|m)$ covers $Q(x\backslash\alpha)$ and $Q(x,k)$ as a special case. All the functions in Elfun18 with k as an argument are defined in this way. However, we note that, from a computational point of view, $Q(x|m)$ reduces the domain of k. For example, in the double precision numerical model domain, $10^{-308} < |k| < 10^{308}$ is reduced to $10^{-154} < |k| < 10^{154}$.

A similar situation occurs with the theta functions. Their argument is nome q, $|q| < 1$, which, in the real case, is given either by $q = \cos(\pi\tau)$, where τ is the half-period ratio, or by $q = \exp\left(-\pi K'(m)/K(m)\right)$. Thus, we have four possibilities:

$$\vartheta(x,q),\ \vartheta(x;\tau),\ \vartheta(x,k),\ \text{ and } \vartheta(x|m).$$

Currently, the collection covers only the arguments q and m.

3. Collection description

Effun18 consists of low-level functions (with scalar arguments) and high-level functions (with matrix arguments). The actual computation is conducted using low-level functions that have the parameter m as the input (m-functions). Most of these computational functions are based on MATLAB translations of the Algol procedures from [7–9], Fortran functions from [10,11], and Pascal procedures from [12]. More precisely, for the computation of elliptic integrals and an inverse of the Jacobian elliptic functions either Bulirsch's integrals *el1*, *el2*, or *el3* or Carlson's integrals *rc*, *rd*, *rg*, or *rj* are used [1]. The core function for the computation of the Jacobian elliptic function is the *sncndn* procedure from [1]. Most of the low-level functions that use module k as an input argument are wrappers, i.e., functions that call appropriate low-level m-functions by setting $m = k^2$.

All high-level functions are wrappers that mimic an elemental behavior of a function by calling one of the functions *ufun1*, *ufun2*, *ufun3*, or *ufun4*. Here, the term *elemental* is borrowed from Fortran, which means that high-level functions may be called using matrix arguments of the same size (any of which can be scalar), in which case a corresponding low-level function is applied element-wise with a conforming matrix return value. All higher-level functions check their input data, namely, the matrix class and a number of arguments.

Naming convention: All high-level functions use Maple-style naming, i.e., the functions begin with an uppercase letter. In this way,

Table A.1
Bulirsch's elliptic integrals.

Function	Matrix arguments	Scalar arguments
$el1\left(x,k'\right) \equiv \int_0^x \frac{dt}{\sqrt{\left(1+t^2\right)\left(1+k'^2t^2\right)}}$	BulirschEL1(X,KC)	el1(x,kc)
$el2\left(x,k',a,b\right) \equiv \int_0^x \frac{\left(a+bt^2\right)dt}{\left(1+t^2\right)\sqrt{\left(1+t^2\right)\left(1+k'^2t^2\right)}}$	BulirschEL2(X,KC,A,B)	el2(x,kc,a,b)
$el3\left(x,k',p\right) \equiv \int_0^x \frac{\left(1+t^2\right)dt}{\left(1+pt^2\right)\sqrt{\left(1+t^2\right)\left(1+k'^2t^2\right)}}$	BulirschEL3(X,KC,P)	el3(x,kc,p)
$cel1\left(k'\right) \equiv el1\left(\infty,k'\right)$	BulirschCEL1(kc)	cel1(kc)
$cel2\left(k',a,b\right) \equiv el2\left(\infty,k',a,b\right)$	BulirschCEL2(KC,A,B)	cel2(kc,a,b)
$cel3\left(k',p\right) \equiv el3\left(\infty,k',p\right)$	BulirschCEL3(KC,P)	cel3(kc,p)
$cel\left(k',p,a,b\right) \equiv \int_0^\infty \frac{\left(a+bt^2\right)dt}{\left(1+pt^2\right)\sqrt{\left(1+t^2\right)\left(1+k'^2t^2\right)}}$	BulirschCEL(KC,P,A,B)	cel(kc,p,a,b)

Table A.2
Carlson's elliptic integrals.

Function	Matrix argument	Scalar arguments
$R_C(x,y) \equiv R_F(x,y,y)$	CarlsonRC(X,Y)	rc(x,y)
$R_D(x,y,z) \equiv R_J(x,y,z,z)$	CarlsonRD(X,Y,Z)	rd(x,y,z)
$R_F(x,y,z) \equiv \frac{1}{2}\int_0^\infty \frac{dt}{\sqrt{(t+x)(t+y)(t+z)}}$	CarlsonRF(X,Y,Z)	rf(x,y,z)
$R_G(x,y,z) \equiv \frac{1}{4}\int_0^\infty \frac{1}{\sqrt{(t+x)(t+y)(t+z)}}\left(\frac{x}{t+x}+\frac{y}{t+y}+\frac{z}{t+z}\right)t\,dt$	CarlsonRG(X,Y,Z)	rg(x,y,z)
$R_J(x,y,z,p) \equiv \frac{3}{2}\int_0^\infty \frac{dt}{(t+p)\sqrt{(t+x)(t+y)(t+z)}}$	CarlsonRJ(X,Y,Z,P)	rj(x,y,z,p)

Table A.3
Incomplete elliptic integrals, $-\infty < m < \infty$.

Jacobi form	Matrix argument	Scalar argument
$\mathbf{B}(x\|m) \equiv \int_0^x \frac{\left(1-t^2\right)dt}{\sqrt{\left(1-t^2\right)\left(1-mt^2\right)}}$	mEllipticB(X,M)	melB(x,m)
$\mathbf{D}(x\|m) \equiv \int_0^x \frac{t^2 dt}{\sqrt{\left(1-t^2\right)\left(1-mt^2\right)}}$	mEllipticD(X,M)	meld(x,m)
$\mathbf{E}(x\|m) \equiv \int_0^x \sqrt{\frac{1-mt^2}{1-t^2}}\,dt$	mEllipticE(X,M)	melE(x,m)
$\mathbf{F}(x\|m) \equiv \int_0^x \frac{dt}{\sqrt{\left(1-t^2\right)\left(1-mt^2\right)}}$	mEllipticF(X,M)	melF(x,m)
$\boldsymbol{\Pi}(x,\nu\|m) \equiv \int_0^x \frac{dt}{\left(1-\nu t^2\right)\sqrt{\left(1-t^2\right)\left(1-mt^2\right)}}$	mEllipticPi(X,NU,M)	melPi(x,m)
Legendre form		
$B(\phi\|m) \int_0^\phi \frac{\cos^2\theta d\theta}{\sqrt{1-m\sin^2\theta}}$	mpEllipticB(PHI,M)	mpelB(phi,m)
$D(\phi\|m) \equiv \int_0^\phi \frac{\sin^2\theta d\theta}{\sqrt{1-m\sin^2\theta}}$	mpEllipticD(PHI,M)	mpelD(phi,m)
$E(\phi\|m) \equiv \int_0^\phi \sqrt{1-m\sin^2\theta}\,d\theta$	mpEllipticE(PHI,M)	mpelE(phi,m)
$F(\phi\|m) \equiv \int_0^\phi \frac{d\theta}{\sqrt{1-m\sin^2\theta}}$	mpEllipticF(PHI,M)	mpelF(phi,m)
$\Pi(\phi,\nu\|m) \equiv \int_0^\phi \frac{d\theta}{\left(1-\nu\sin^2\theta\right)\sqrt{1-m\sin^2\theta}}$	mpEllipticPi(PHI,NU,M)	mpelPi(phi,nu,m)
Jacobi's second form		
$\varepsilon(u\|m) \equiv \int_0^u \mathrm{dn}^2(t\|m)\ dt$	mJacobiEpsilon(U,M)	mjepsilon(u,m)
$\Lambda(u,\nu\|m) \equiv \int_0^u \frac{dt}{1-\nu\,\mathrm{sn}^2(t\|m)}$	mJacobiLambda(U,NU,M)	mjlambda(u,nu,m)

Table A.4
Complete elliptic integrals, $-\infty < m \le 1$.

Definition	Matrix argument	Scalar argument
$B(m) \equiv \mathbf{B}(1\|m) = B(\pi/2\|m)$	mEllipticB(M)	melB(m)
$C(m) \equiv \int_0^1 \frac{t^2\left(1-t^2\right)dt}{\sqrt{\left(1-t^2\right)\left(1-mt^2\right)^3}} = \int_0^{\pi/2} \frac{\sin^2\theta\cos^2\theta d\theta}{\sqrt{\left(1-m\sin^2\theta\right)^3}}$	mEllipticC(M)	melC(m)
$D(m) \equiv \mathbf{D}(1\|m) = D(\pi/2\|m)$	mEllipticD(M)	melD(m)
$E(m) \equiv \mathbf{E}(1\|m) = E(\pi/2\|m) = \varepsilon(K(m)\|m)$	mEllipticE(M)	melE(m)
$K(m) \equiv \mathbf{F}(1\|m) = F(\pi/2\|m)$	mEllipticK(M)	melK(m)
$\Pi(\nu\|m) \equiv \boldsymbol{\Pi}(1,\nu\|m) = \Pi(\pi/2,\nu\|m) = \Lambda(K(m),\nu\|m)$	mEllipticPi(NU,M)	melPi(nu,m)

a mismatch cannot occur with MATLAB SMT functions, which begin with a lowercase letter. Each function is available with either the modulus k or the parameter m as an argument. In the latter case, the function name starts with the letter "m". Incomplete elliptic integrals are given either in Jacobi form, Legendre form, or Jacobi's second form. The Legendre form begins with the letter "p" (from the argument "phi"). The low-level inverse functions begin with the letter "i, " a high-level inverse function with the word "Inverse".

The names of some functions are not standard. Thus, we call Jacobi's second form of elliptic integral $\varepsilon(u|m)$ as JacobiEpsilon. Also, for convenience, we named the functions $\Lambda(u,\nu|m)$ and $\Omega(\phi,\nu|m)$ JacobiLambda and JacobiOmega, although a Jacobi does not use these functions [13].

Table A.5
Complementary complete elliptic integrals, $0 \le m < \infty$.

Definition	Matrix arguments	Scalar arguments
$E'(m) \equiv E(1-m)$	mEllipticCE(M)	melCE(m)
$K'(m) \equiv K(1-m)$	mEllipticCK(M)	melCK(m)
$\Pi'(\nu\|m) \equiv \Pi(\nu\|1-m)$	mEllipticCPi(NU,M)	melCPi(nu,m)

We enhanced the above description using some examples. If the modulus k is applied as an argument, then the low-level name

Table A.6
Related functions, $-\infty < m \leq 1$.

Definition	Matrix arguments	Scalar arguments
$\mathbf{Z}(u\|m) \equiv \varepsilon(u\|m) - \frac{E(m)}{K(m)}u$	mJacobiZeta(U,M)	mjzeta(u,m)
$Z(\phi\|m) \equiv E(\phi\|m) - \frac{E(m)}{K(m)}F(\phi\|m)$	mpJacobiZeta(PHI,M)	mpjzeta(phi,m)
$\boldsymbol{\Omega}(u,\nu\|m) \equiv \boldsymbol{\Pi}(u,\nu\|m) - \frac{\Pi(\nu\|m)}{K(m)}u$	mJacobiOmega(U,NU,M)	mjomega(u,nu,m)
$\Omega(\phi,\nu\|m) \equiv \Pi(\phi,\nu\|m) - \frac{\Pi(\nu\|m)}{K(m)}F(\phi\|m)$	mpJacobiOmega(PHI,NU,M)	mpjomega(phi,nu,m)
$\Lambda_0(\beta\|m) \equiv \frac{m' \sin(2\beta)\Pi\left(\frac{m}{1-m'\sin^2\beta}\|m\right)}{\pi\sqrt{1-m'\sin^2\beta}}$	mHeumanLambda(BETA,M)	mhlambda(beta,m)

Table A.7
Jacobian elliptic functions, $-\infty < m < \infty$.

Definition	Matrix arguments	Scalar arguments
$\mathrm{am}(x\|m) \equiv \int_0^x \mathrm{dn}(t\|m)\,dt$	mJacobiAM(X,M)	mjam(x,m)jam(x,k)
$\mathrm{cd}(x\|m) \equiv \mathrm{cn}(x\|m)/\mathrm{dn}(x\|m)$	mJacobiCD(X,M)	mjcd(x,m)
$\mathrm{cn}(x\|m) \equiv \cos(\mathrm{am}(x\|m))$	mJacobiCN(X,M)	mjcn(x,m)
$\mathrm{cs}(x\|m) \equiv \mathrm{cn}(x\|m)/\mathrm{ds}(x\|m)$	mJacobiCS(X,M)	mjcs(x,m)
$\mathrm{dc}(x\|m) \equiv \mathrm{dn}(x\|m)/\mathrm{cn}(x\|m)$	mJacobiDC(X,M)	mjdc(x,m)
$\mathrm{dn}(x\|m) \equiv \sqrt{1-m\sin^2(\mathrm{am}(x\|m))}$	mJacobiDN(X,M)	mjdn(x,m)
$\mathrm{ds}(x\|m) \equiv \mathrm{dn}(x\|m)/\mathrm{sn}(x\|m)$	mJacobiDS(X,M)	mjds(x,m)
$\mathrm{nc}(x\|m) \equiv 1/\mathrm{cn}(x\|m)$	mJacobiNC(X,M)	mjnc(x,m)
$\mathrm{nd}(x\|m) \equiv 1/\mathrm{dn}(x\|m)$	mJacobiND(X,M)	mjnd(x,m)
$\mathrm{ns}(x\|m) \equiv 1/\mathrm{sn}(x\|m)$	mJacobiNS(X,M)	mjns(x,m)
$\mathrm{sc}(x\|m) \equiv \mathrm{sn}(x\|m)/\mathrm{cn}(x\|m)$	mJacobiSC(X,M)	mjsc(x,m)
$\mathrm{sd}(x\|m) \equiv \mathrm{sn}(x\|m)/\mathrm{dn}(x\|m)$	mJacobiSD(X,M)	mjsd(x,m)
$\mathrm{sn}(x\|m) \equiv \sin(\mathrm{am}(x\|m))$	mJacobiSN(X,M)	mjsn(x,m)

Table A.8
Inverse of Jacobian elliptic functions, $-\infty < m < \infty$.

Definition	Matrix arguments	Scalar arguments
$\mathrm{am}^{-1}(x\|m) \equiv \mathbf{F}(x\|m)$	mInverseJacobiAM(X,M)	mijam(x,m)
$\mathrm{cd}^{-1}(x\|m) \equiv \mathbf{F}\left(\sqrt{\frac{1-x^2}{1-mx^2}}\|m\right)$	mInverseJacobiCD(X,M)	mijcd(x,m)
$\mathrm{cn}^{-1}(x\|m) \equiv \mathbf{F}\left(\sqrt{1-x^2}\|m\right)$	mInverseJacobiCN(X,M)	mijcn(x,m)
$\mathrm{cs}^{-1}(x\|m) \equiv \mathbf{F}\left(\frac{1}{\sqrt{1+x^2}}\|m\right)$	mInverseJacobiCS(X,M)	mijcs(x,m)
$\mathrm{dc}^{-1}(x\|m) \equiv \mathbf{F}\left(\sqrt{\frac{x^2-1}{x^2-m}}\|m\right)$	mInverseJacobiDC(X,M)	mijdc(x,m)
$\mathrm{dn}^{-1}(x\|m) \equiv \mathbf{F}\left(\sqrt{\frac{1-x^2}{m}}\|m\right)$	mInverseJacobiDN(X,M)	mijdn(x,m)
$\mathrm{ds}^{-1}(x\|m) \equiv \mathbf{F}\left(\frac{1}{\sqrt{m+x^2}}\|m\right)$	mInverseJacobiDS(X,M)	mijds(x,m)
$\mathrm{nc}^{-1}(x\|m) \equiv \mathbf{F}\left(\sqrt{1-\frac{1}{x^2}}\|m\right)$	mInverseJacobiNC(X,M)	mijnc(x,m)
$\mathrm{nd}^{-1}(x\|m) \equiv \mathbf{F}\left(\sqrt{\frac{x^2-1}{mx^2}}\|m\right)$	mInverseJacobiND(X,M)	mijnd(x,m)
$\mathrm{ns}^{-1}(x\|m) \equiv \mathbf{F}\left(\frac{1}{x}\|m\right)$	mInverseJacobiNS(X,M)	mijns(x,m)
$\mathrm{sc}^{-1}(x\|m) \equiv \mathbf{F}\left(\frac{1}{\sqrt{1+x^2}}\|m\right)$	mInverseJacobiSC(X,M)	mijsc(x,m)
$\mathrm{sd}^{-1}(x\|m) \equiv \mathbf{F}\left(\frac{x}{\sqrt{1+x^2}}\|m\right)$	mInverseJacobiSD(X,M)	mijsd(x,m)
$\mathrm{sn}^{-1}(x\|m) \equiv \mathbf{F}(x\|m)$	mInverseJacobiSN(X,M)	mijsn(x,m)

for the Jacobian elliptic function *sn* is "jsn", and its higher level name is "JacobiSN". If the parameter m is used as an argument, then the names are "mjsn" and "mJacobiSN". For the inverse function sn^{-1}, the names are "ijsn" and "InverseJacobiSN", and "mijsn" and "mInverseJacobiSN". As a numerical example, we calculate $\mathrm{sn}(0.5, 0.5)$ and $\mathrm{sn}(0.5|0.5)$ along with their inversion. We then have

jsn(0.5,0.5) = 0.475082936028536,
ijsn(0.475082936028536,0.5) = 0.500000000000000
mjsn(0.5,0.5) = 0.470750473655657, and
mijsn(0.470750473655657,0.5) = 0.500000000000000.

Table A.9
Lemniscate functions.

Definition	Matrix argument	Scalar argument
$\mathrm{cl}(x) \equiv \frac{\sqrt{2}}{2}\mathrm{cn}\left(x\sqrt{2}, \frac{\sqrt{2}}{2}\right)$	GaussCL(X)	gcl(x)
$\mathrm{sl}(x) \equiv \frac{\sqrt{2}}{2}\mathrm{sd}\left(x\sqrt{2}, \frac{\sqrt{2}}{2}\right)$	GaussSL(X)	gsl(x)
$\mathrm{cl}^{-1}(x) \equiv \int_x^1 \frac{dt}{\sqrt{1-t^2}}$	InverseGaussCL(X)	igcl(x)
$\mathrm{sl}^{-1}(x) \equiv \int_0^x \frac{dt}{\sqrt{1-t^2}}$	InverseGaussSL(X)	Igsl(x)

Table A.10
Gudermannian functions.

Definition	Matrix argument	Scalar argument
$\mathrm{gd}(x) \equiv \int_0^x \frac{dt}{\cosh t} = \tan^{-1}(\sinh x)$	GudermannGD	gd(x)
$\mathrm{gd}^{-1}(x) \equiv \int_0^x \frac{dt}{\cos t} = \sinh^{-1}(\tan x)$	InverseGuderman-nGD(X)	igd(x)

The low-level name for a function that calculates the Jacobi form of an elliptic integral of the first kind with k as the argument is "elF", and the name of the higher-level function is "EllipticF". If the parameter m is used as an argument, then the names are "melF" and "mEllipticF". If the Legendre form of an integral is used, then the names are either "pelf" and "pEllipticF" or "mpelF" and "mpEllipticF". As a numerical example, we calculate $F(0.5, 0.5)$, $F(0.5|0.5)$, $\mathbf{F}(0.5, 0.5)$, and $\mathbf{F}(0.5|0.5)$. We then have the following:

pelF(0.5,0.5) = 0.505088727578648
mpelF(0.5,0.5) = 0.510467135628005
elF(0.5,0.5) = 0.529428627051906
melF(0.5,0.5) = 0.535622732805403

4. Performance

4.1. Correctness

From the functions in the collection, we expect that, for real arguments, they return either a real number (including Inf) or NaN. Therefore, any computational function checks whether its actual arguments belong to the domain where the function values are real numbers. If an argument falls out of the domain the function returns NaN without warning. It is thus the user's responsibility to check the obtained results. For example, the Jacobi form of an elliptic integral has real values when $1 - x^2 \geq 0$ and $1 - mx^2 \geq 0$. From this, it follows that when $m \leq 1$ then $|x| \leq 1$, and if $m > 1$ then $|x| < 1/\sqrt{m}$. The parameter m is thus any real number, whereas the domain of x depends on m. For the second example, we consider the complete elliptic integral $\mathbf{K}(m)$. In the function melK, its domain is set to $-\infty < m \leq 1$ because for $m > 1$ the values of the integral are complex.

Table A.11
Jacobi theta functions, $0 \le q \le 1$.

Function	Matrix argument	Scalar argument
$\vartheta_1(x,q) \equiv 2q^{1/4}\sum_{n=1}^{\infty}(-1)^n q^{n(n+1)}\sin[(2n+1)x]$	JacobiTheta1(X,Q)	jtheta1(x,q)
$\vartheta_2(x,q) \equiv 2q^{1/4}\sum_{n=1}^{\infty} q^{n(n+1)}\cos[(2n+1)x]$	JacobiTheta2(X,Q)	jtheta2(x,q)
$\vartheta_3(x,q) \equiv 1+2\sum_{n=1}^{\infty} q^{n^2}\cos(2nx)$	JacobiTheta3(X,Q)	jtheta3(x,q)
$\vartheta_4(x,q) \equiv 1+2\sum_{n=1}^{\infty}(-1)^n q^{n^2}\cos(2nx)$	JacobiTheta4(X,Q)	jtheta4(x,q)

Table A.12
Neville theta functions, $0 \le q \le 1$, $0 \le m \le 1$.

Function	Matrix arguments	Scalar arguments
$\theta_c(x,q) \equiv \theta_2\left(\frac{\pi x}{2K(k(q))}, q\right)/\theta_2(0,q)$	NevilleThetaC(X,Q)	nthetac(x,q)
$\theta_d(x,q) \equiv \theta_3\left(\frac{\pi x}{2K(k(q))}, q\right)/\theta_3(0,q)$	NevilleThetaD(X,Q)	nthetad(x,q)
$\theta_n(x,q) \equiv \theta_4\left(\frac{\pi x}{2K(k(q))}, q\right)/\theta_4(0,q)$	NevilleThetaN(X,Q)	nthetan(x,q)
$\theta_s(x,q) \equiv \frac{2K(k)}{\pi}\theta_1\left(\frac{\pi x}{2K(k(q))}, q\right)/\theta_1'(0,q)$	NevilleThetaS(X,Q)	nthetas(x,q)
$\theta_c(x\vert m) \equiv \theta_c\left(\frac{\pi x}{2K(m)}, q(m)\right)$	mNevilleThetaC(X,M)	mnthetac(x,m)
$\theta_d(x\vert m) \equiv \theta_d\left(\frac{\pi x}{2K(m)}, q(m)\right)$	mNevilleThetaD(X,M)	mnthetad(x,m)
$\theta_n(x\vert m) \equiv \theta_n\left(\frac{\pi x}{2K(m)}, q(m)\right)$	mNevilleThetaN(X,M)	mnthetan(x,m)
$\theta_s(x\vert m) \equiv \theta_s\left(\frac{\pi x}{2K(m)}, q(m)\right)$	mNevilleThetaS(X,M)	mnthetas(x,m)

Table A.13
Related functions.

Function	Matrix arguments	Scalar arguments
$\text{nome}(k) \equiv \exp\left(-\frac{\pi K'(k)}{K(k)}\right)$	EllipticNome(X,K)	elnome(k)
$\text{nome}(k) = q$	InverseEllipticNome(Q)	ielnome(q)

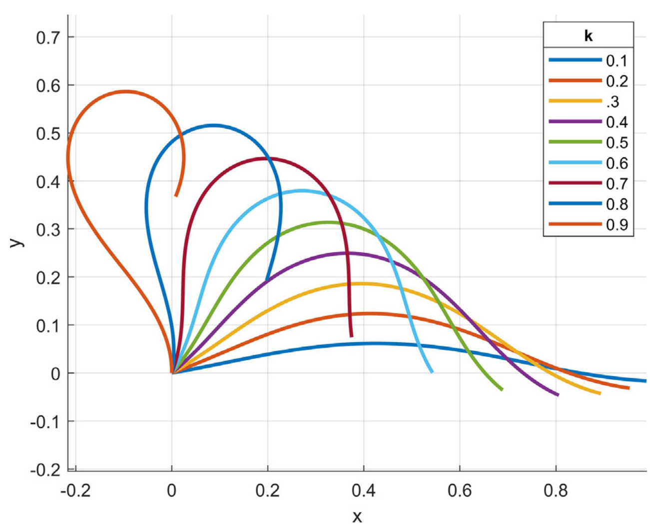

Fig. 2. Flexural elastica for $\omega = 5$, $C = 1$, and different values of k.

Some care was also taken to design the functions that intercept their poles and their limits for infinite arguments. Thus, for example, $K(1) = \infty$ and $K(-\infty) = 0$. Therefore, for these arguments, the collection function melK provides melK(1) = Inf and melK(−Inf) = 0. This approach differs from typical implementations where NaN is returned if $m = -\infty$, for example.

To confirm that, for $m = k^2$, the m-functions provide the same value as the k-functions, all k-functions are defined by (3), i.e., all k-functions are wrappers. Also, if a function $f(x)$ is even or odd then for $x < 0$ it is evaluated as $f(-x) = f(x)$ or $f(-x) = -f(x)$.

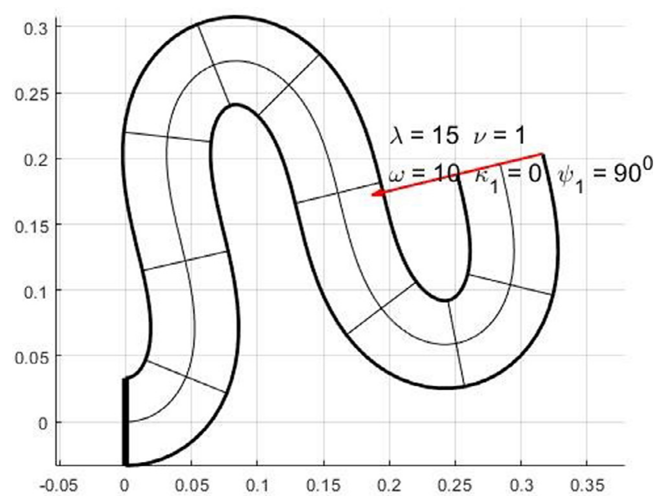

Fig. 3. Shearless cantilever under the follower force. The length of the deformed base curve 0.78555198, which agrees with the value given in Table 1 [14].

4.2. Robustness

In the present collection, the number and type of input data are checked in higher-level routines. Low-level routines intercept inputs that are NaN and out of the function domain.

All computational routines were tested for extremely small and large arguments. Some additional checks are added to the function, which use successive Landen transformations to prevent them from hanging if any variable becomes NaN.

4.3. Efficiency

As previously stated, the functions in the collection mimic the elemental-behavior of the matrix arguments, i.e., the code is not vectorized. This has an impact on the function efficiency if a vector or matrix argument is used. From Tables 1–3, we can see that vectorized MATLAB functions are on a matrix argument two-to five-times faster than the present functions. However, this is not critical because, as we can see from Table 1, the computation time for a million function evolutions is about 0.5 s. In addition, from Tables 1 and 3, we can also see that the present functions are faster than MATLAB functions when a function is called with a scalar argument. As expected, the present functions are much faster than elliptic functions from the MATLAB symbolic toolbox. All computations were conducted on HP Z820 workstation.

4.4. Accuracy

The accuracy of the functions in the collection was tested in two ways: through a comparison with MATLAB elliptic functions from the SMT, and by testing various functional identities. Examples results of these tests are shown in Tables 4 and 5.

5. Examples

As a simple application of the present formulas, consider Euler's flexural elastica, which has the following parametric form [14,15]:

$$
\begin{aligned}
x(u) &= \frac{2}{\omega}\left[\varepsilon(\omega s + C, k) - \varepsilon(C, k)\right] - s,\\
y(s) &= \frac{2k}{\omega}\left[\operatorname{cn}(C, k) - \operatorname{cn}(\omega s + C, k)\right]
\end{aligned} \tag{8}
$$

where ω is a load parameter and C is a constant. The following program produces the shapes of the elastica shown in Fig. 2.

```
% Flexural elastica

% Data
omega = 5;              % load parameter
cc    = 1;              % integration constant
kk    = 0.1:0.1:0.9;    % eliptic modulus

% Plot shapes

figure
hold on
s = 0:0.01:1;
C = cc*ones(size(s));
for k = kk
    x = -s + 2*(JacobiEpsilon(omega*s + C,k)
       - JacobiEpsilon(C,k))/omega;
    y = 2*k*(JacobiCN(C,k) - JacobiCN
       (omega*s + C,k))/omega;
    plot(x,y,'LineWidth',2)
end
h=legend('0.1','0.2','.3','0.4','0.5','0.6','0.7',
'0.8','0.9',... 'Location','best');
title(h,'k')
xlabel('x')
ylabel('y')
axis equal
grid on
hold off
```

As a second example, we consider a finite-strain elastic cantilever under the follower force from [14]. The coordinates X, Y of the deformed cantilever base curve are

$$
X = x\cos\alpha + y\sin\alpha, \quad Y = -x\sin\alpha + y\cos\alpha \tag{9}
$$

where

$$
\begin{aligned}
x &= -\left[\frac{(1-\nu)\,\omega^2}{2\lambda^2} + \frac{\widetilde{\omega}^2}{\omega^2}\right]s + \frac{2\widetilde{\omega}}{\omega^2}\left\{\varepsilon(\widetilde{\omega}s + C, k) - \varepsilon(C, k)\right.\\
&\quad - m^2\left[\frac{\operatorname{sn}(\widetilde{\omega}s + C, \widetilde{k})\operatorname{cn}(\widetilde{\omega}s + C, \widetilde{k})\operatorname{dn}(\widetilde{\omega}s + C, \widetilde{k})}{1 + m^2\operatorname{cn}^2(\widetilde{\omega}s + C, \widetilde{k})}\right.\\
&\quad \left.\left. - \frac{\operatorname{sn}(C, \widetilde{k})\operatorname{cn}(C, \widetilde{k})\operatorname{dn}(C, \widetilde{k})}{1 + m^2\operatorname{cn}^2(C, \widetilde{k})}\right]\right\}\\
y &= \frac{2k\widetilde{\omega}\sqrt{1+m^2}}{\omega^2}\left[\frac{\operatorname{cn}(C, \widetilde{k})}{1 + m^2\operatorname{cn}^2(C, \widetilde{k})} - \frac{\operatorname{cn}(\widetilde{\omega}s + C, \widetilde{k})}{1 + m^2\operatorname{cn}^2(\widetilde{\omega}s + C, \widetilde{k})}\right]
\end{aligned} \tag{10}
$$

and

$$
\begin{aligned}
&\widetilde{\omega} \equiv \omega\sqrt{1 + \frac{\nu\omega^2}{\lambda^2}\left(2k^2 - 1\right)}, \quad m^2 \equiv \frac{\frac{\nu\omega^2}{\lambda^2}k^2}{1 - \frac{\nu\omega^2}{\lambda^2}\left(1 - k^2\right)}\\
&\widetilde{k}^2 \equiv \frac{k^2 + m^2}{1 + m^2}
\end{aligned} \tag{11}
$$

The non-dimensional parameters in the above equations are the load parameter ω, generalized slenderness ratio λ, and stiffness ratio ν. For the cantilever under the follower force, we have

$$
k^2 = \sin^2\frac{\psi_1}{2} \tag{12}
$$

$$
\alpha = 2\sin^{-1}\left(k\frac{\operatorname{sn}(C, \widetilde{k})}{\sqrt{1 + m^2\operatorname{cn}^2(C, \widetilde{k})}}\right) \tag{13}
$$

$$
C = -\widetilde{\omega} + K(\widetilde{k}) \tag{14}
$$

where ψ_1 is given as the clockwise angle between the direction of the force and inward normal to the cantilever deformed cross-section at the free end. When the cantilever is shearless, i.e., when $\nu = 1$, its deformed length L is given by

$$
\begin{aligned}
L &= 1 - \left(1 + \frac{2k^2}{m^2}\right)\frac{\omega^2}{\lambda^2} + \frac{2k^2}{m^2\widetilde{\omega}}\frac{\omega^2}{\lambda^2}\\
&\quad\times\left[\Lambda\left(\widetilde{\omega} + C, \frac{m^2}{1+m^2}, \widetilde{k}\right) - \Lambda\left(C, \frac{m^2}{1+m^2}, \widetilde{k}\right)\right]
\end{aligned} \tag{15}
$$

The program used to implement these formulas is given in Appendix B. The shape of the deformed cantilever shown in Fig. 3 matches the shape in Fig. 2c [14].

6. Conclusion

The described collection contains a complete set of MATLAB functions for a calculation of the real elliptic integrals and real Jacobian elliptic functions in all their possible forms. The functions were tested for their correctness, robustness, efficiency, and accuracy. The collection is freely available from https://www.mathworks.com/matlabcentral/fileexchange/65915-elfun18. The reference manual with examples is available from https://www.researchgate.net/publication/323399074_elfun18_-_Elliptic_Integrals_Jacobian_Elliptic_Functions_and_Theta_Functions_-_Reference_Manual_v11.

Declaration of competing interest

The authors whose names are listed immediately below certify that they have NO affiliations with or involvement in any organization or entity with any financial interest (such as honoraria; educational grants; participation in speakers bureaus; membership, employment, consultancies, stock ownership, or other equity interest; and expert testimony or patent-licensing arrangements), or non-financial interest (such as personal or professional relationships, affiliations, knowledge or beliefs) in the subject matter or materials discussed in this manuscript.

Appendix A. Contents of the collection

See Tables A.1–A.13.

Appendix B. Program for calculation of the shape of the finite-strain cantilever under the follower force

```
close all
% Data
psi1   = pi/3;  % angle between the direction of force an inward normal at free end
lambda = 10;    % generalized slenderness > 0
nu     = -1;    % stiffness ratio [-1..1]
omega  = 4;     % load factor
cantilever( psi1, lambda, nu, omega)

function cantilever( psi1, lambda, nu, omega)
% Cantilever under follower force ([1] Sec 5.1)
%
% Reference:
%  [1]  M.Batista - A closed-form solution for Reissner planar finite-strain
%       beam using Jacobi elliptic functions, International Journal of Solids
%       and Structures 87 (2016) 153-166

% Calculate constants
eta2   = (omega/lambda)^2;
k      = sin(psi1/2);                                        % Eq 53
m2     = nu*eta2*k^2/(1 - nu*eta2*(1 - k^2));                % Eq 34a
omega1 = omega*sqrt(1 + nu*eta2*(2*k^2 - 1));                % Eq 34b
k1     = sqrt((k^2 + m2)/(1 + m2));                          % Eq 39
C0     = -omega1 + elK(k1);                                  % Eq 67
alpha  = 2*asin(k*jsn(C0,k1)/sqrt(1 + m2*jcn(C0,k1)^2));     % Eq 65

% Calculate shape
[ x, y, phi ] = shape( 0:0.01:1 );

% Calculate length of deformed beam
if nu == 1  % shearless beam only  Eq 46
    L = 1 - (1 + 2*k^2/m2)*eta2 + 2*k^2/m2/omega1*eta2*...
        (jlambda(omega1 + C0,m2/(1 + m2),k1) - jlambda(C0,m2/(1 + m2),k1));
else
    L = NaN;
end
fprintf('%s = %.16g\n','C',     C0)
fprintf('%s = %.16g\n','alpha', alpha)
fprintf('%s = %.16g\n','L',     L)
fprintf('%s = %.16g\n','X', x(end))
fprintf('%s = %.16g\n','Y', y(end))

% Plot shape
figure
hold on
% center line
plot(x,y,'k')
s = -1:0.1:1;
h = 1/lambda/2; % aux. beam thickness
% plot outer lines
plot(x + h*sin(phi),y - h*cos(phi),'k','LineWidth',2)
plot(x - h*sin(phi),y + h*cos(phi),'k','LineWidth',2)
% plot section
for n = 1:9
    [x0,y0,phi0] = shape(0.1*n);
    plot(x0 + h*sin(phi0)*s,y0 - h*cos(phi0)*s,'k');
end
% plot end sections
[x0,y0,phi0] = shape(0);
plot(x0 + h*sin(phi0)*s,y0 - h*cos(phi0)*s,'k','LineWidth',4);
[x0,y0,phi0] = shape(1);
plot(x0 + h*sin(phi0)*s,y0 - h*cos(phi0)*s,'k');
% plot force
```

```
s = -2:0.1:2;
quiver(x0 + h*sin(phi0),y0 - h*cos(phi0), -h*sin(phi0), h*cos(phi0),...
    4,'r','Linewidth',1.5);
txt = sprintf('%s = %g  %s = %g\n%s = %g  %s = %g  %s = %g%s',...
    '\lambda',lambda,'\nu',nu,'\omega',omega,'\kappa',kappa1,...
    '\psi',psi1*180/pi,'^[3]')
text(0.2,0.2,txt,'FontSize',14);
axis equal
grid on
hold off

    function [ x, y, phi ] = shape( s )
        C = C0*ones(size(s));
        xx = ((nu - 1)*omega^2*s/lambda^2)/ 2 + 2*omega1/(omega^2)*...
            ((EllipticE(k1) / EllipticK(k1) - 1/2)*omega1*s + ...
            JacobiZeta(omega1*s + C, k1) - JacobiZeta(C, k1) - ...
            (m2*(JacobiSN(omega1*s + C, k1).*JacobiCN(omega1*s + C, k1).*...
            JacobiDN(omega1*s + C, k1)./(1 + m2*JacobiCN(omega1*s + C, k1).^ 2)...
            - JacobiSN(C, k1).* JacobiCN(C, k1).*JacobiDN(C, k1)./...
            (1 + m2*JacobiCN(C, k1).^2))));     % Eq 43a
        yy = -(2*omega1*k*sqrt(1 + m2)/omega^2)*(JacobiCN(omega1*s + C,k1)./...
            (1 + m2*JacobiCN(omega1*s + C, k1).^ 2) -  JacobiCN(C, k1)./...
            (1 + m2*JacobiCN(C, k1).^ 2));      % Eq 43b
        x  =  xx*cos(alpha) + yy*sin(alpha);    % Eq 19a
        y  = -xx*sin(alpha) + yy*cos(alpha);    % Eq 19b
        phi = 2*asin(k*JacobiSN(omega1*s + C,k1)./...
            sqrt(1 + m2*JacobiCN(omega1*s + C,k1).^2)) - alpha;  % Eq 41, 22
    end

end
```

References

[1] Carlson BC. In: Olver FWJ, National Institute of Standards and Technology (US), editors. Elliptic Integrals. New York: Cambridge University Press : NIST, Cambridge; 2010, p. xv, 951 p.

[2] Jahnke E, Emde F. Tables of functions with formulae and curves. 4th ed. New York: Dover Publications; 1945.

[3] Abramowitz M, Stegun IA. Handbook of mathematical functions: with formulas, graphs, and mathematical tables. Mineola, N.Y.: Dover; 1972.

[4] Milne-Thomson LM. JacobiAn elliptic functions and theta functions. In: Abramowitz M, Stegun IA, editors. Handbook of mathematical functions: with formulas, graphs, and mathematical tables. Mineola, N.Y.: Dover; 1972.

[5] Reinhardt WP, Walker PL. JacobiAn elliptic functions. In: Olver FWJ, editor. NIST handbook of mathematical functions. New York: Cambridge University Press: NIST, Cambridge; 2010, p. xv, 951 p.

[6] Byrd PF, Friedman MD. Handbook of elliptic integrals for engineers and scientists. 2nd ed. Berlin, New York: Springer-Verlag; 1971.

[7] Bulirsch R. Numerical calculation of elliptic integrals and elliptic functions. Numer Math 1965;7:78–90.

[8] Bulirsch R. Numerical calculation of elliptic integrals and elliptic functions. II. Numer Math 1965;7:353–4.

[9] Bulirsch R. Numerical calculation of elliptic integrals and elliptic functions. III. Numer Math 1969;13:305–15.

[10] Carlson BC, Notis EM. Algorithm 577 - algorithms for incomplete elliptic integrals [S21]. ACM Trans Math Software 1981;7:398–403.

[11] Press WH. Numerical Recipes in Fortran 77 the Art of Scientific Computing. 2nd ed. Cambridge: Cambridge University Press; 2006.

[12] Ehrhardt W. The AMath and DMath special functions, in. 2018.

[13] Jacobi KGJ. Fundamenta nova theoriae functionum ellipticarum. Regiomonti: Bornträger; 1829.

[14] Batista M. A closed-form solution for reissner planar finite-strain beam using Jacobi elliptic functions. Int J Solids Struct 2016;87:153–66.

[15] Love AEH. A treatise on the mathematical theory of elasticity. 4th ed. New York: Dover Publications; 1944.