

Nonlinear Analysis of Flexible Beams Undergoing Large Rotations Via Symbolic Computations

XIAOFENG YUAN ^a and YUKIO TADA ^{b,*}

^a*Division of System Science, Graduate School of Science and Technology,
Kobe University, Kobe, Japan;* ^b*Department of Computer and Systems
Engineering, Faculty of Engineering, Kobe University, Kobe, Japan*

(Received 3 October 2000)

In this paper, a two-stage approach is presented for analyzing flexible beams undergoing large rotations. In the first stage, the symbolic forms of equations of motion and the Jacobian matrix are generated by means of MATLAB and written into a MATLAB script file automatically, where the flexible beams are described by the unified formulation presented in our previous paper. In the second stage, the derived equations of motion are solved by means of implicit numerical methods. Several comparison computations are performed. The two-stage approach proves to be much more efficient than pure numerical one.

Keywords: Differential equation; Direct integration; Symbolic computation; FEM; Geometric nonlinearity; Inertia nonlinearity; Matlab

1. INTRODUCTION

There are many practical structures which have beam-like shapes and undergo large overall motions, such as satellite antennas, helicopter blades, robot arms, *etc.* Many works have been done in formulation of flexible beams undergoing large rotations. Yoo [1] used a non-Cartesian variable along with two Cartesian variables to describe the small elastic deformation of a flexible straight beam, with shear

*Corresponding author. e-mail: tada@cs.kobe-u.ac.jp

and rotary effects being neglected. Valembois [2] compared several methods for modelling flexible beams in multibody systems, such as finite segment method, finite element method and assumed mode method. Pedersen [3] described the system only by the position of the nodes in the inertial frame which yielded a constant mass matrix. It resulted in a faster numerical integration, but the formulation was restricted to isoparametric shape functions which had only translational degree of freedoms. So beam elements were not applicable.

For short and stubby beams, the effects of rotary inertia and shear deformation become important, especially when higher modes of vibration are considered, and their neglect can lead to significant errors. On the other hand, for slender beams, it will cause large errors in response analysis to neglect the second geometric stiffness matrices and nonlinear forces induced by elastic deformations when large deflections in elements and/or large axial forces are involved. In Bakr's formulation [4] the effects of shear deformation are considered. And in Mayo's formulation [5] the higher order stiffness matrices and higher order nonlinear forces induced by elastic deformations are considered.

As for numerical integration methods, implicit methods such as BDF(Backward Differentiation Formulas) method and implicit Adams method are frequently used. Andrzejewski and his co-workers gave a survey [6].

There are two different kinds of formalisms, the numerical and the symbolical ones. The numerical equations of motion have to be generated for each time step of the integration code and for each parameter variation. The symbolical equations, however, need to be generated only once, so they are especially helpful for real time applications and parameter optimization [7]. Symbolical formalism for the dynamic analysis of rigid multibody systems has been successfully used for more than a decade. But applications to flexible multibody systems are very few [8]. Nevertheless, the drawbacks in symbolic computing, in particular for the flexible multibody system are that only simple structural elements can be described by means of a symbolic approach for the displacement field. Furthermore, a large number of degrees of freedom may result in long expressions, which may be difficult to handle by a computer algebra system.

Finite difference approximation of nonlinear ordinary equations results in nonlinear algebraic equations. Jacobian is essential for the

solution of the algebraic equations. Normally, the Jacobian is approximated numerically, which results in a lot of additional evaluations of the right hand side term of the state equations. Furthermore, in each evaluation of the right hand side term, the system matrices such as mass matrix M , damping matrix C and stiffness matrix K and force vector F should be assembled from the element level. It is very time-consuming.

In our previous paper [9], a unified formulation was presented and the pure numerical formalism was adopted. In the formulation, up to the second order geometric stiffening effects, as well as the rotary and shear effects were considered, therefore it was suitable for various beams, from stubby ones to very slender ones. Nonlinearity in this formulation arises from two sources. These are namely geometric elastic nonlinearity and inertia nonlinearity. Geometric nonlinearity is the result of retaining the quadratic terms in the strain-displacement relationship. This results in the classical geometric stiffness matrix, the second geometric stiffness matrix and the nonlinear elastic forces, with rotary inertia and shear deformation effects being considered. On the other hand, inertia nonlinearity arises from the gross rotations of the beams in which the reference motion and elastic deformation are coupled.

In present paper, a two-stage approach for analyzing the flexible beams is proposed. The first stage is to generate the symbolic forms of equations of motion and the Jacobian matrix by symbolic toolbox of MATLAB5.3 from the unified formulation and to write them into a Matlab script file *OdeF.m* automatically. The second one is to solve the derived ODE by means of implicit numerical methods. This approach, may be mentioned as a symbolic numerical approach, proves to be much more efficient than pure numerical one and is suitable for real time computation. For the completeness, the unified formulation is summarized here.

2. FORMULATION OF ROTATING FLEXIBLE BEAMS

2.1. Coordinate Systems

The configuration of the object system is determined by three coordinate systems, that is, inertia coordinate system XY , floating reference frame X_1Y_1 which is fixed at the beam under undeformed

configuration and shares the same rigid rotation as the beam, and elemental coordinate system $(X_2Y_2$, for example) which rotates simultaneously with X_1Y_1 . U_{x2} and U_{y2} in Figure 1 are the elastic displacements in X_2 and Y_2 directions of an arbitrary point P , respectively.

The generalized coordinates of the beam are given by

$$y = [\theta \quad q^T]^T \tag{1}$$

where θ is the rigid body rotation, and q is elastic node displacement vector.

To consider the rotary inertia and shear deformation effects, the following shape function [10] is used for Timoshenko beam elements.

$$N = \frac{1}{1+\phi} \begin{bmatrix} (1-\xi)(1+\phi) & 6(\xi-\xi^2)\eta & [-1+4\xi-3\xi^2-(1-\xi)\phi]l\eta \\ 0 & 1-3\xi^2+2\xi^3+(1-\xi)\phi & [\xi-2\xi^2+\xi^3+(1/2)(\xi-\xi^2)\phi]l \\ \xi(1+\phi) & 6(-\xi+\xi^2)\eta & (2\xi-3\xi^3-\xi\phi)l\eta \\ 0 & 3\xi^2-2\xi^3+\xi\phi & [-\xi^2+\xi^3-(1/2)(\xi-\xi^2)\phi]l \end{bmatrix} \tag{2}$$

where l is the length of the beam element, $\xi = x/l$, $\eta = y/l$ and ϕ is the shear deformation parameter which is given by

$$\phi = 12EI/(\kappa GA^*l^2) \tag{3}$$

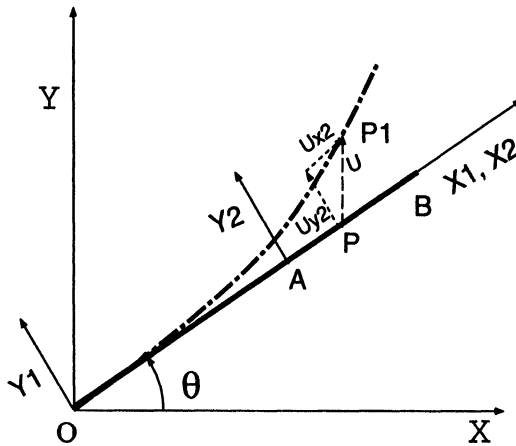


FIGURE 1 A beam undergoing large rotation.

where κ is the shear coefficient, G is the shear modulus, A^* is the cross sectional area, and EI is the flexural rigidity of the beam.

2.2. Kinetic Energy

Kinetic energy of the beam is expressed as follows:

$$T = \sum_i^{n_e} \frac{1}{2} \int_{V^i} \rho^i \dot{r}^{iT} \dot{r}^i dV^i = \begin{Bmatrix} \theta \\ q \end{Bmatrix}^T \begin{bmatrix} M_{\theta\theta}(y) & M_{\theta q}(y) \\ M_{\theta q}^T(y) & M_{qq} \end{bmatrix} \begin{Bmatrix} \theta \\ q \end{Bmatrix} \quad (4)$$

where n_e is the number of elements, ρ^i and r^i are the mass density and the position of arbitrary point in element i , respectively. The submatrix M_{qq} is constant, while $M_{\theta\theta}(y)$ and $M_{\theta q}(y)$ are time-variable. Refer to Bakr and Shabana [10] for the explicit expression of element mass matrix.

2.3. Strain Energy

Strain tensor for finite strain deformation is

$$\varepsilon_{ik} = \varepsilon_{ik}^l + \varepsilon_{ik}^{nl} = \frac{1}{2} \left(\frac{\partial u_i}{\partial x_k} + \frac{\partial u_k}{\partial x_i} + \frac{\partial u_i}{\partial x_j} \frac{\partial u_j}{\partial x_k} \right) \quad (5)$$

where ε_{ik}^l and ε_{ik}^{nl} are linear part and nonlinear part of the strain, respectively. The strain energy expression U of a Timoshenko beam can be written as follows [4]:

$$U = \frac{E}{2} \int \varepsilon_{xx}^2 dV + 2G \int \varepsilon_{xy}^2 dV \quad (6)$$

Using Eq. (5) and neglecting small terms, one obtains

$$U = U_E + U_S + U_{nl11} + U_{nl12} + U_{nl2} \quad (7)$$

where U_E , U_S , U_{nl11} , U_{nl12} and U_{nl2} are linear axial, linear shear, classic geometric elastic nonlinear, shear geometric and higher order geometric elastic nonlinear components of the strain energy,

respectively. These components are expressed as:

$$U_E = \frac{E}{2} \int \left(\frac{\partial u_x}{\partial x} \right)^2 dV \quad (8)$$

$$U_S = \frac{G}{2} \int \left[\left(\frac{\partial u_x}{\partial y} \right)^2 + \left(\frac{\partial u_y}{\partial x} \right)^2 + 2 \left(\frac{\partial u_x}{\partial y} \frac{\partial u_y}{\partial x} \right) \right] dV \quad (9)$$

$$U_{nl11} = \frac{E}{2} \int \frac{\partial u_x}{\partial x} \left(\frac{\partial u_y}{\partial x} \right)^2 dV \quad (10)$$

$$U_{nl12} = G \int \frac{\partial u_x}{\partial x} \left[\left(\frac{\partial u_x}{\partial y} \right)^2 + \left(\frac{\partial u_x}{\partial y} \frac{\partial u_y}{\partial x} \right) \right] dV \quad (11)$$

$$U_{nl2} = \frac{E}{8} \int \left(\frac{\partial u_y}{\partial x} \right)^4 dV \quad (12)$$

U is different from that of Bakr [4] in existing the last term, and from that of Mayo [5] in considering the rotary and shear effects.

Using the shape function of Eq. (2) and summing over the elements of the whole beam, the strain energy expression can be re-written as:

$$U = \frac{1}{2} \mathbf{q}^T \mathbf{K} \mathbf{q} \quad (13)$$

where the system stiffness matrix is

$$\mathbf{K} = \mathbf{K}_E + \mathbf{K}_S + \mathbf{K}_{nl11} + \mathbf{K}_{nl12} + \mathbf{K}_{nl2} \quad (14)$$

where \mathbf{K}_E and \mathbf{K}_S are constant matrices, while \mathbf{K}_{nl11} and \mathbf{K}_{nl12} are composed of linear terms of \mathbf{q} , and \mathbf{K}_{nl2} is composed of the second order terms of \mathbf{q} .

Mayo [5] indicated that those formulations which do not include the fourth order terms of strain energy (*i.e.*, the second geometric stiffness matrix and the nonlinear elastic forces) may lead to wrong responses in applications involving large deflections in the elements and/or large axial forces.

After applying Lagrange's equations, the equations of motion for a flexible rotating beam can be given by

$$M(\mathbf{y})\ddot{\mathbf{y}} + (C + G(\mathbf{y}))\dot{\mathbf{y}} + K(\mathbf{y})\mathbf{y} = F(\mathbf{y}, t) \quad (15)$$

$$F(\mathbf{y}, t) = F_{ex}(\mathbf{y}, t) + Q_v(\mathbf{y}) + Q_g(\mathbf{y}) + Q_h(\mathbf{y}) \quad (16)$$

where $F_{ex}(\mathbf{y}, t)$ is the generalized external force vector, $G(\mathbf{y})$ is gyroscopic matrix which is skew-symmetric, $Q_v(\mathbf{y})$ arises from the variability of the mass matrix, $Q_g(\mathbf{y})$ and $Q_h(\mathbf{y})$ are nonlinear elastic forces induced by the elastic deformations. The former is the axial nodal force vector which is composed of the second order terms of \mathbf{y} , and the latter includes transverse force and moment of nodes, and is composed of the third order terms of \mathbf{y} .

We added the terms reflecting the effect of shear deformation into Mayo's formulation, and corrected the errors existing in the expressions of higher order stiffness matrix and nonlinear elastic forces in Mayo's paper.

3. NONLINEAR ANALYSIS APPROACHES

3.1. Numerical Formalism

After transforming the equations of motion into state equations, methods for the 1st order ODE can be applied. In the present work, several solvers in the ODE suite [11] are adopted.

$$M_{eq}(\mathbf{v})\dot{\mathbf{v}} = F_{eq}(\mathbf{v}, t) \quad (17)$$

where,

$$M_{eq}(\mathbf{v}) = \begin{bmatrix} M(\mathbf{y}) & \mathbf{0} \\ \mathbf{0} & I \end{bmatrix}; \quad \mathbf{v} = \begin{bmatrix} \mathbf{p} \\ \mathbf{y} \end{bmatrix} \quad (18)$$

$$F_{eq}(\mathbf{v}, t) = \begin{bmatrix} F(\mathbf{y}, t) - (C + G(\mathbf{y}))\mathbf{p} - K(\mathbf{y})\mathbf{y} \\ \mathbf{p} \end{bmatrix} \quad (19)$$

The following steps should be performed for each function evaluation:

- (1) Compute M_{eq} and F_{eq}
- (2) Decompose $M = LDL^T$ by Cholesky method
- (3) Solve $\dot{\mathbf{v}} = \begin{bmatrix} (LDL^T)^{-1} & 0 \\ 0 & \dot{\mathbf{I}} \end{bmatrix} F_{eq}$

When rigid motion is prescribed, the governing ODE will be reduced to that with a constant mass matrix, and the step (2) above can be performed only once in advance.

The numerical formalism is described as Figure 2(a). The state equations should be generated from the element level for each evaluation of $\dot{\mathbf{v}}$. In addition, the numerical computations of the Jacobian matrix of the equations result in increase of evaluation number of $\dot{\mathbf{v}}$.

3.2. Symbolic Formalism

In symbolic formalism (see Fig. 2(b)), the whole problem can be divided into two independent stages, *i.e.*, the generation of the model and the solution of the equations of motion. The state equations and the Jacobian matrix need to be generated only once.

In the first stage, we use symbolical tool box of Matlab5.3 to generate the equations of motion in the form of state equations and the Jacobian, and to output them to a Matlab subroutine. In the second stage, we adopt the implicit solvers *ode15s*, *ode23t* and *ode23tb*

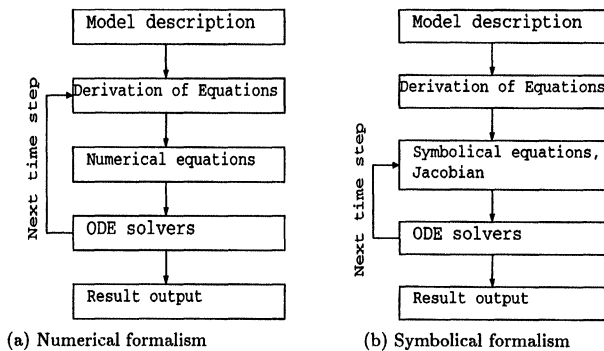


FIGURE 2 Comparison of two formalisms.

in the ODE suite to solve the governing ODE. The *ode15s* is a variable order solver with the order from low to medium, which includes both BDF method and the Numerical Differentiation Formulas, or NDF method. The *ode23t* is an implementation of the trapezoidal method of order 2 and is suitable for moderately stiff problem, while *ode23tb* is an implicit Runge–Kutta formula with a first stage that is a trapezoidal rule step and a second stage that is a BDF of order two. Because the equations and Jacobian matrix are given explicitly, the solution process becomes much more efficient.

4. EXAMPLES

Let's consider a slender beam, rotating around the hinge with the following given function of angle-time relationship.

$$\theta = \begin{cases} (\omega_s/t_s)((t^2/2) + (t_s/2\pi)^2(\cos(2\pi t/t_s) - 1)) & \text{for } t < t_s \\ \omega_s(t - (t_s/2)) & \text{for } t \geq t_s \end{cases} \quad (20)$$

where steady-state angular velocity $\omega_s = 6\text{m/s}$ and spin-up time $t_s = 15\text{sec}$. Relative parameters are shown in Table I.

4.1. Comparison of Numerical and Symbolical Formalisms

Table II compares efficiencies of the following three approaches in the cases of $DOF = 18$ and $DOF = 48$:

- (1) pure numerical approach (*num*)
- (2) symbolical-numerical approach without explicit Jacobian (*symb1*)
- (3) symbolical-numerical approach with explicit Jacobian (*symb2*)

It is shown that *symb2* is the most efficient in solution of the governing ODE, much faster than *num*, although it takes long time to

TABLE I Relative parameters of the example

Lengths of beams	10 m	Elastic modulus	$7 \times 10^{10} \text{N/m}^2$
Area of cross section	$4 \times 10^{-4} \text{m}^2$	Linear density	1.2 kg/m
Area moment of inertia	$2 \times 10^{-7} \text{m}^4$	Damping matrix <i>C</i>	neglected

TABLE II Comparison for the three approaches

Items	Case of $DOF=18$			Case of $DOF=48$		
	<i>num</i>	<i>symb1</i>	<i>symb2</i>	<i>num</i>	<i>symb1</i>	<i>symb2</i>
time for deriving Ode	–	184	1165	–	1553	39398
time for solving Ode	643	27	9	54678	54	23
successful steps	349	114	114	5147	98	98
failed attempts	229	27	27	3792	22	22
function evaluations	5336	408	225	182555	269	173
partial derivatives	114	5	5	1712	1	1
LU decompositions	314	52	52	5255	43	43
solving linear systems	1117	222	223	16490	171	171

derive the ODE. The *symb1* takes the shortest total time for derivation and solution of the ODE. The derivation of Jacobian is time-consuming. In the pure numerical approach, much more time steps and function evaluations are necessary, much more failed attempts happened, and each function evaluation implies more computations than that of symbolic approaches. Table II says that the ratios of the CPU times for solving the governing ODE by the three approaches are 71.4:3:1 when the degree of freedom of the problem is 18, and 2377.3:2.3:1 when the degree of freedom becomes 48. Obviously, as the scale of the problem increases, the symbolical-numerical approaches become much more and more efficient than the pure numerical one in solution of the governing ODE. Of course, when using symbolical-numerical approach, the scale of the problem is limited by the algebraic computation ability under the available computer conditions. Approach *sym2* is suitable for real computation, *sym1* could be used when pursuing shortest total CPU time, while *num* is suitable for large scale problems in which symbolic softwares can not work.

4.2. Comparison of Solutions of 3 Approximate Models

The solutions of the following three approximate models are compared as shown in Figure 3.

model 1: Constant stiffness matrix

model 2: Up to the first order geometric stiffness matrix

model 3: Up to the second order geometric stiffness matrix and nonlinear elastic forces

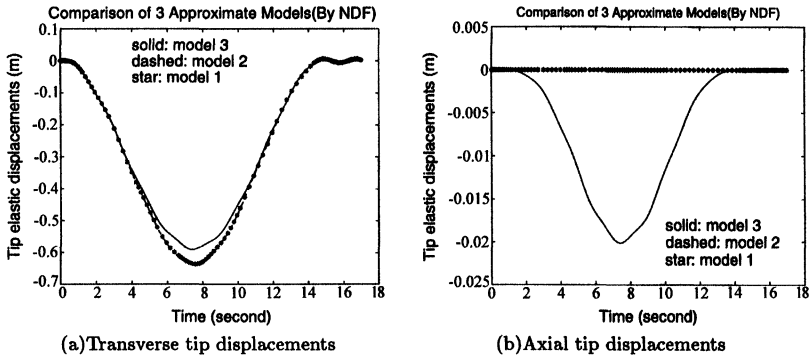


FIGURE 3 Tip displacement of 3 approximate models.

Figure 3 describes the tip elastic displacements in transverse and axial directions of the three models by the NDF method, for the case with the length of 10 m. It shows that, for a slender beam, results of the last model are apparently different from (actually, better than) those of the first two models.

4.3. Comparison of Efficiencies of 4 Solvers

Comparison of computation costs of 4 different solvers (see Tab. IV) is performed with the same parameters shown in Table III, where N_e is the number of elements, L is the length of the beam, $tspan$ is the time

TABLE III Parameters for computation

N_e	L	$tspan$	$order$	$RelTol$	$AbsTol$
4	10 m	[0, 15.2]	2nd	0.001	0.000001

TABLE IV Comparison of efficiencies of 4 solvers

Methods	BDF	NDF	ode23t	ode23tb
successful steps	911	757	1864	706
failed attempts	383	328	259	512
function evaluations	6080	5841	5173	10547
partial derivatives	146	149	50	272
LU decompositions	584	496	835	800
solving linear systems	2429	2115	3922	4460
CPU time (s)	1293	1149	1019	2042

interval, *RelTol* is the relative error tolerance and *AbsTol* is the absolute error tolerance. The results are obtained by the pure numerical approach.

Table IV shows that the implementation of the trapezoidal method(*ode23t*) is the most efficient for the problem. And the next is the NDF method. It may mean that the problem is mildly stiff.

5. CONCLUSIONS

In this paper, a symbolic-numerical approach is presented. Symbolic toolbox of MATLAB5.3 is used to derive the equations of motion and Jacobian from the unified formulation. Then implicit numerical methods are used to solve the governing ODE. The approach proves to be much more efficient than the pure numerical approach.

References

- [1] Yoo, H. H., Ryan, R. R. and Scott, R. A. (1994). Dynamics of flexible beams undergoing overall motion, *Journal of Sound and Vibration*, **181**(2), 261–278.
- [2] Valembois, R. E., Fiset, P. and Samin, J. C. (1997). Comparison of various techniques for modelling beams in multibody dynamics, *Nonlinear Dynamics*, **12**, 367–397.
- [3] Pedersen, N. L. (1997). On the formulation of flexible multibody systems with constant mass matrix, *Multibody System Dynamics*, **1**(3), 323–337.
- [4] Bakr, E. M. (1996). Dynamic analysis of geometrically nonlinear robot manipulators, *Nonlinear Dynamics*, **11**, 329–346.
- [5] Mayo, J. and Dominguez, J. (1996). Geometrically non-linear formation of flexible multibody systems in terms of beam elements: geometric stiffness, *Computers & Structures*, **59**(6), 1039–1050.
- [6] Andrzejewski, Th., Bock, H. G., Eich, E. and von Schwerin, R., Recent advances in the numerical integration of multibody systems, Schiehlen, W. (Ed.), *Advanced Multibody System Dynamics*, Kluwer Academic Publishers, 1993, pp. 127–151.
- [7] Schiehlen, W. (1997). Multibody system dynamics: roots and perspectives, *Multibody System Dynamics*, **1**, 149–188.
- [8] Melzer, F. (1996). Symbolic computations in flexible multibody systems, *Nonlinear Dynamics*, **9**, 147–164.
- [9] Yuan, X. and Tada, Y. (1999). Nonlinear analysis of flexible beams undergoing large rotations, *Memoirs of the Graduate School of Science and Technology*, Kobe University, **17A**, 71–79.
- [10] Bakr, E. M. and Shabana, A. A. (1987). Timoshenko beams and flexible multibody system dynamics, *Journal of Sound and Vibration*, **116**(1), 89–107.
- [11] Shampine, L. F. and Reichelt, M. W. (1997). The Matlab ODE suite, *SIAM J. Sci. Comp.*, **18**, 1–22.