AIAA SciTech Forum
9 - 13 January 2017, Grapevine, Texas
55th AIAA Aerospace Sciences Meeting

10.2514/6.2017-1671

Adjoint-based derivative evaluation methods for flexible multibody systems with rotorcraft applications

Komahan Boopathy* and Graeme J. Kennedy†
Georgia Institute of Technology, Atlanta, GA, 30332-0150, USA.

High-fidelity rotorcraft simulations involve tight coupling between aerodynamics, structures and structural dynamics. As a step towards developing fully-integrated high-fidelity rotorcraft simulation and gradient-based design optimization tool, this paper presents analysis and discrete-adjoint sensitivity formulations for flexible multibody systems. The governing equations are treated in a descriptor form, as second-order implicit functions of state and design variables. The solution procedures for the governing equations are discussed in the context of different time marching schemes, including backward difference formulas, diagonally implicit Runge–Kutta, Newmark and Adams–Bashforth–Moulton methods. The corresponding time-dependent discrete adjoint derivatives are formulated and numerically verified with the complex-step method. The assessments of parallel scalability of the implementation are shown along with flexible multibody dynamics applications of the framework.

Nomenclature

f	objective function	scalar	$\ddot{\mathbf{q}}$	second time derivative of states	m-vector
h	step size	scalar	$\mathbf{R}$	governing equations	m-vector
$\mathscr{L}$	Lagrangian	scalar	$\mathbf{S}, \mathbf{T}$	state approximation equations	m-vector
m	number of state variables	scalar	t	time	scalar
n	number of design variables	scalar	$\mathbf{x}$	design variables	n-vector
N	number of time intervals	scalar	α, β, γ	time marching coefficients	scalar
p	order of accuracy	scalar	λ, ψ, ϕ	adjoint variables	m-vector
$\mathbf{q}$	state variables	m-vector	*Subscripts*		
$\dot{\mathbf{q}}$	first time derivative of states	m-vector	k	time step index	scalar

I. Introduction and Background

High-fidelity rotorcraft simulations involve tight coupling between multiple disciplines including aerodynamics, structures, and structural dynamics. Gradient-based design optimization techniques offer the potential to help identify design modifications necessary to achieve the ever more demanding rotorcraft performance goals, such as higher forward flight speeds, higher operational altitudes and longer endurance limits. This work takes a step towards fully-integrated high-fidelity rotorcraft simulation and gradient-based design optimization framework. The focus is on the structural and structural dynamics disciplines and development of a simulation tool with discretely consistent adjoint-based derivative evaluation techniques, for time accurate flexible multibody dynamic simulations. The objective of this work is to formulate numerical solution procedures for the analysis of flexible multibody systems, as well as the corresponding time-dependent discrete-adjoint sensitivities. In the remainder of this section, we provide a brief review of solution and derivative evaluation approaches for flexible multibody systems. Section II contains the development of the time marching schemes for second-order systems and the derivation of the associated adjoint equations. Section III presents the numerical verification of the adjoint derivatives and applications to flexible multibody dynamics. Finally, Section IV concludes the paper summarizing key findings.

A. Governing Equations and Functionals of Interest

Flexible multibody systems are governed by second-order differential equations in time. There are several approaches to derive the equations of motion, where each approach differs in the choice of coordinates and the treatment of kinematic constraints, resulting in a different form for the governing system of equations [6]. For example, using the Newton–Euler approach, the system is broken down into individual components, and the forces and moments on

*Graduate Student, School of Aerospace Engineering, 270 Ferst Drive, komahan@gatech.edu, AIAA Student Member.

†Assistant Professor, School of Aerospace Engineering, 270 Ferst Drive, graeme.kennedy@aerospace.gatech.edu, AIAA Member.

each body are used to obtain the equations of motion. When unconstrained generalized coordinates are employed to parametrize the system, the resulting equations of motion are a set of ordinary differential equations (ODEs). However, when constrained generalized coordinates are used, the equations of motion are a set of differential algebraic equations (DAEs). Another approach is to use analytical dynamics techniques to derive the governing equations. In these methods, which include the methods of Lagrange, Hamilton, Maggi and others (see Refs. [6, 31] for details), the system is treated as a whole and scalar quantities, such as the kinetic and potential energy, are used to derive the equations of motion. A central issue when formulating the equations of motion is the treatment of kinematic constraints, that enforce compatibility requirements on the system. While these constraints can be easily eliminated for simple systems, for complex systems it is simpler to retain the kinematic constraints. Once the equations of motion are derived, they are often converted into a canonical first-order form, known as the *state-space representation*, and several well-known numerical solution procedures exist for such first-order systems [8, 18, 19]. The first-order conversion process can also lead to problems when the mass matrix is singular. Furthermore, the physics can become obscured due to the substitutions carried out during the conversion. In this paper, we use the descriptor form of the governing equations of motion which can be written as follows

$$\mathbf{R}(\ddot{\mathbf{q}}, \dot{\mathbf{q}}, \mathbf{q}, \mathbf{x}, t) = 0, \tag{1}$$

where $\mathbf{q}$ are the state variables, $\dot{\mathbf{q}}$ and $\ddot{\mathbf{q}}$ are the first and second time derivatives of the state variables, respectively, and $\mathbf{x}$ is the vector of design variables. Note that the vector $\mathbf{q}$ contains position coordinates, rotational parametrization variables and the Lagrange multipliers associated with the kinematic constraints. The descriptor form allows for the treatment of different formulations of the governing equations under a common mathematical framework. We refrain from converting the governing equations into an equivalent first-order form, and instead use the descriptor form directly. In the context of the design of flexible multibody systems, it is necessary to define objective and constraint functionals of interest. We work with a generic functional of interest, defined as follows

$$f(\mathbf{x}) = \int_0^T F(\ddot{\mathbf{q}}, \dot{\mathbf{q}}, \mathbf{q}, \mathbf{x}, t)\, dt, \tag{2}$$

where F is the integrand which depends on the state variables and their time derivatives as well as the design variables. Note that the integral (2) can be used to evaluate aggregation functionals such as p-norm and Kreisselmeier–Steinhauser (KS) [26, 28], which smoothly approximate the maximum value of a quantity of interest (e.g. stress, strain). The functional form (2) is used in the development of the adjoint method used to evaluate the derivative of the functional of interest with respect to the design variables.

B. Time Marching Schemes

Since general closed-form solutions of the governing equations (1) are not available, it is necessary to numerically integrate the the governing equations forward in time for a given set of initial conditions $\mathbf{q}_0$ and $\dot{\mathbf{q}}_0$. The mathematical properties of DAEs present additional challenges compared to that of ODEs. As a result, not all numerical methods, that are well-suited for ODEs, can be applied to DAEs. One approach to solve DAEs is to split the governing equations into stiff and non-stiff parts, and to solve the stiff algebraic part using implicit integration schemes, and the non-stiff part using explicit methods. Several numerical methods have been developed for first-order stiff and non-stiff systems [8, 19]. In this paper, we focus on implicit time marching techniques, that are well-suited to the integration of numerically stiff systems of DAEs, governing the motion of flexible multibody dynamics. While implicit integration methods require the solution of a nonlinear system of equations at each time step, this additional computational cost is offset by the improved stability properties and the resulting larger time steps enabled by implicit methods.

C. Derivative Evaluation Methods

A number of methods exist to compute the derivatives of outputs from numerical simulations governed by differential equations.

1. Numerical Methods

The finite-difference method is the simplest numerical method to compute derivatives, however, its accuracy is strongly influenced by the choice of the step size and its computational cost increases rapidly with the number of design variables. The complex-step method [29, 37], on the other hand, provides highly accurate derivative estimates, since it does not suffer from subtractive cancellations. However, the complex-step method is computationally more expensive compared to the finite-difference method due to the use of complex number arithmetic. Automatic differentiation

(AD) [36], which is based on the direct application of the chain rule of differentiation to the simulation source code, provides machine accurate derivatives, but the computational cost and ease of implementation are not optimal in most cases.

2. The Adjoint Method

The adjoint method is a technique for obtaining the derivative of a functional output of interest with respect to many input variables in a computationally efficient manner. It has been successfully applied to structural [1, 2, 7, 22, 27], aerodynamic [4, 9, 25] and coupled aeroelastic simulations [21, 30]. The adjoint method requires the solution of a linear system governing the adjoint variables, for each functional output of interest. The computational cost of computing the derivative of the functional of interest using the adjoint method is nearly independent of the number of design variables. The adjoint equations can be derived either using a continuous adjoint formulation [4], or from the discrete governing equations [9]. Various authors have developed adjoint methods for flexible multibody dynamics applications since the late 1990s [12, 15, 16, 33, 35]. These methods are specific to different explicit formulations of governing equations and state-space representations. In this work, the adjoint is derived directly for the discrete governing equations in second-order descriptor form using different time marching schemes.

II. Time Marching and Discrete-Adjoint For Second-Order Systems

This section presents the methods for numerical solution of the governing equations and the development of the associated adjoint equations from the descriptor form. The time marching methods presented within this section are

1. Backward Difference Formulas (BDF) [14, 23],
2. Newmark method [17, 34],
3. Adams–Bashforth–Moulton method (ABM) [5, 32],
4. Diagonally Implicit Runge–Kutta (DIRK) [3, 13].

These time marching schemes use the state variable values from different time steps to evaluate the subsequent values of state variables, starting from a given set of initial conditions $\mathbf{q}_0$ and $\dot{\mathbf{q}}_0$. The subscript, k, on the state variables refer to the corresponding time value, t_k. In this work, a constant step size, h, is employed for time marching. The coefficients of time marching, that are used for forming the weighted linear combinations of the state variables at each time step, are denoted as α, β and γ. The choice of coefficients influence the order of accuracy and stability properties of the time marching schemes.

A. Backward Difference Formulas (BDF)

The BDF method was first proposed by Curtiss [14] and Henrici [23]. The BDF method is an implicit multistep method based on finite differences: the higher-order difference operators are obtained by repetitive application of the first-order difference operator. As a result, the first-derivative approximation requires state variable values at $p+1$ points, while the second derivative approximation requires state variables at $2p+1$ points, where p is the order of accuracy. The BDF method is suitable for the solution of stiff ODEs and DAEs and several solution packages, such as `ODEPACK/LSODE` [24], and `DASSL` [8], employ this method.

1. Solution of the State Variables

State Approximations: The primary unknowns of the BDF time marching scheme are the state variables, $\mathbf{q}_k$, at each time step, k. The first and second time derivatives of the state variables, $\dot{\mathbf{q}}_k$, and $\ddot{\mathbf{q}}_k$, are obtained using

$$\begin{aligned} \dot{\mathbf{q}}_k &= \frac{1}{h}\sum_{i=0}^{p} \alpha_i \mathbf{q}_{k-i} + \mathscr{O}(h^p), \\ \ddot{\mathbf{q}}_k &= \frac{1}{h^2}\sum_{i=0}^{2p} \beta_i \mathbf{q}_{k-i} + \mathscr{O}(h^p). \end{aligned} \tag{3}$$

The coefficients α_i and β_i depend on the order of approximation, p. The first and second time derivatives of the state variables are linear combinations of the state variables scaled with BDF coefficients, as illustrated in Figure 1.

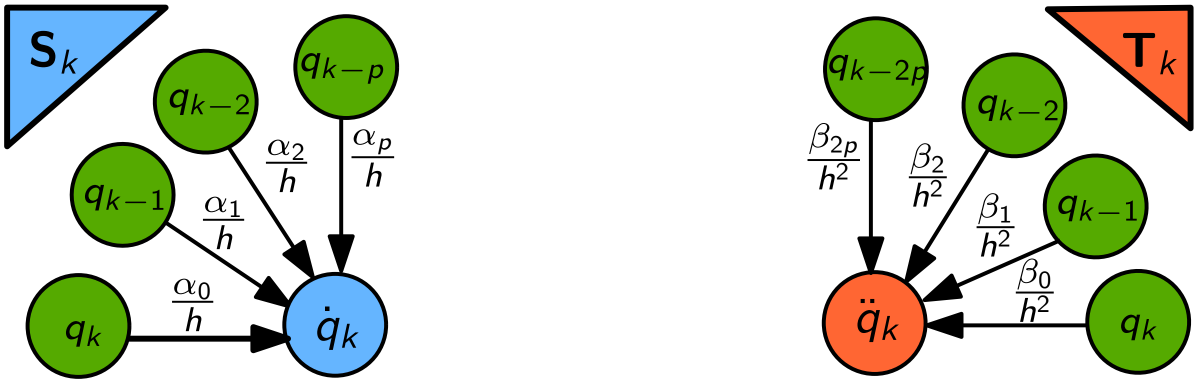


Figure 1: A weighted linear combination of the state variables with scaled BDF coefficients yielding the first (left) and second time derivatives of state variables (right), denoted as state approximation equations, $\mathbf{S}_k$ and $\mathbf{T}_k$, respectively.

SOLUTION OF THE NONLINEAR SYSTEM: Once the time derivatives of states have been approximated using Equation (3), the implicit system of nonlinear equations at each time step becomes

$$\mathbf{R}_k(\ddot{\mathbf{q}}_k, \dot{\mathbf{q}}_k, \mathbf{q}_k, \mathbf{x}, t_k) = 0. \tag{4}$$

This nonlinear system of equations is solved iteratively using Newton's method. The iterative updates to the unknown variables are obtained by solving a linearization of the governing equations with respect to the primary unknowns, $\mathbf{q}_k$, as follows

$$\left[\frac{\beta_0}{h^2}\frac{\partial \mathbf{R}}{\partial \ddot{\mathbf{q}}} + \frac{\alpha_0}{h}\frac{\partial \mathbf{R}}{\partial \dot{\mathbf{q}}} + \frac{\partial \mathbf{R}}{\partial \mathbf{q}}\right] \Delta\mathbf{q}_k = -\mathbf{R}_k(\ddot{\mathbf{q}}_k, \dot{\mathbf{q}}_k, \mathbf{q}_k, \mathbf{x}, t_k). \tag{5}$$

The secondary and tertiary updates to the first and second time derivatives of the state variables are readily obtained by scaling the state variable update, $\Delta\mathbf{q}_k$, using the BDF coefficients, at each iteration n of the nonlinear solution. The resulting update formulas to the state variables are

$$\begin{aligned}
\mathbf{q}_k^{n+1} &= \mathbf{q}_k^n + \Delta\mathbf{q}_k^n, \\
\dot{\mathbf{q}}_k^{n+1} &= \dot{\mathbf{q}}_k^n + \frac{\alpha_0}{h}\Delta\mathbf{q}_k^n, \\
\ddot{\mathbf{q}}_k^{n+1} &= \ddot{\mathbf{q}}_k^n + \frac{\beta_0}{h^2}\Delta\mathbf{q}_k^n.
\end{aligned} \tag{6}$$

The use of the secondary and tertiary updates in Equation (6) is preferred since the original backwards difference formulas (3) typically requires more vector operations. The iterative updates to the state variables and their first and second time derivatives are continued until the governing equations are solved to a specified tolerance. The accuracy of adjoint derivatives rely on the accuracy of the solution of the governing equations. Therefore it is important that the discrete nonlinear system (4) is solved to a tight tolerance.

2. *Solution of the Adjoint Variables*

FORMATION OF THE LAGRANGIAN: The adjoint equations are derived using a Lagrangian formulation. The time integral of the functional of interest is discretized as follows,

$$f(\mathbf{x}) = \int_0^T F(\ddot{\mathbf{q}}, \dot{\mathbf{q}}, \mathbf{q}, \mathbf{x}, t)\, \mathrm{d}t \approx \sum_{k=0}^{N} hF_k(\ddot{\mathbf{q}}_k, \dot{\mathbf{q}}_k, \mathbf{q}_k, \mathbf{x}, t_k). \tag{7}$$

The inner product of the governing equations with the adjoint variables, λ, is approximated as follows

$$\int_0^T \lambda^T \mathbf{R}(\ddot{\mathbf{q}}, \dot{\mathbf{q}}, \mathbf{q}, \mathbf{x}, t)\, \mathrm{d}t \approx \sum_{k=0}^{N} h\lambda_k^T \mathbf{R}_k(\ddot{\mathbf{q}}_k, \dot{\mathbf{q}}_k, \mathbf{q}_k, \mathbf{x}, t_k). \tag{8}$$

The state approximation residuals of the BDF method are introduced as follows

$$\begin{aligned}\mathbf{S}_k &= \frac{1}{h}\sum_{i=0}^{p}\alpha_i\mathbf{q}_{k-i} - \dot{\mathbf{q}}_k,\\ \mathbf{T}_k &= \frac{1}{h^2}\sum_{i=0}^{2p}\beta_i\mathbf{q}_{k-i} - \ddot{\mathbf{q}}_k.\end{aligned} \tag{9}$$

The adjoint variables ψ_k and ϕ_k are associated with state approximation residuals $\mathbf{S}_k$ and $\mathbf{T}_k$, respectively. With these definitions, the Lagrangian is defined as the following linear combination:

$$\mathscr{L} = \sum_{k=0}^{N} hF_k + \sum_{k=0}^{N} h\lambda_k^T\mathbf{R}_k + \sum_{k=0}^{N}\psi_k^T\mathbf{S}_k + \sum_{k=0}^{N}\phi_k^T\mathbf{T}_k. \tag{10}$$

The adjoint variables λ_k, ψ_k and ϕ_k, are the unknown weights in the linear combination (10). Once they are determined the total derivative is readily available as a linear combination involving the same weights. The formation of Lagrangian is illustrated in Figure 2 along with corresponding inputs to the equations $\mathbf{R}_k$, $\mathbf{S}_k$, $\mathbf{T}_k$ and F_k. The similarity in inputs to the governing equations $\mathbf{R}_k$, and the functional, F_k are due to their identical mathematical forms.

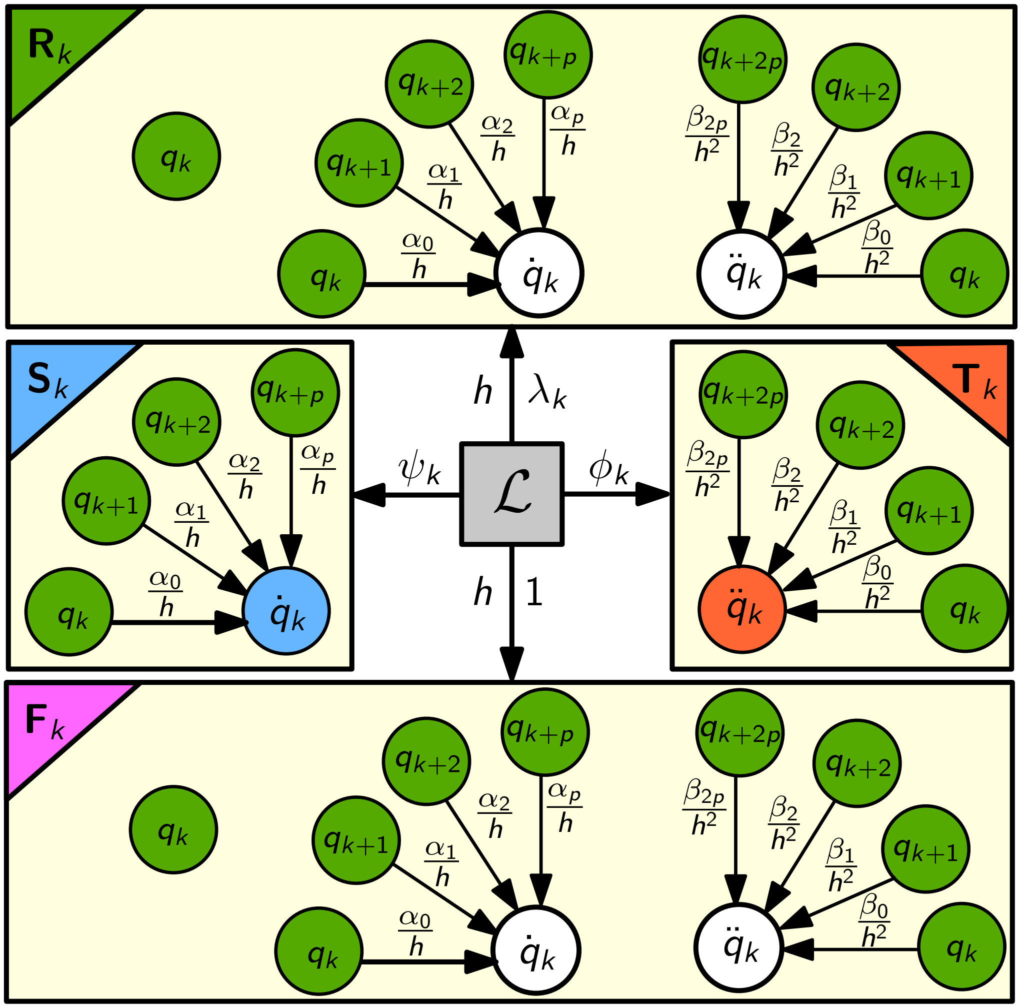


Figure 2: An illustration of the weighted linear combination of equations with corresponding adjoint variables forming the Lagrangian for the BDF method.

THE ADJOINT EQUATIONS: The system of equations to solve for the adjoint variables is obtained from the stationary points of the Lagrangian with respect to the state variables and their first and second time derivatives at each time step. The number of terms in the adjoint system of equations as a function of the order of the system is listed in Table 1.

Table 1: The number of terms in the adjoint system of equations for p-th order BDF method.

	T	**S**	**R**	F	Total
ϕ_k	1				1
ψ_k		1			1
λ_k	$2p+1$	$p+1$	$3p+3$	$3p+3$	$9p+8$

1. Equation for ϕ_k:

The equations to solve for ϕ_k are obtained using $\partial\mathscr{L}/\partial\ddot{\mathbf{q}}_k = 0$. It follows that $\frac{\partial \mathbf{T}_k}{\partial \ddot{\mathbf{q}}_k}^T \phi_k = 0$ which simplifies further to $\phi_k = 0$.

2. Equation for ψ_k:

The equations to solve for ψ_k are obtained using $\partial\mathscr{L}/\partial\dot{\mathbf{q}}_k = 0$. It follows that $\frac{\partial \mathbf{S}_k}{\partial \dot{\mathbf{q}}_k}^T \psi_k = 0$, which simplifies to $\psi_k = 0$.

3. Equation for λ_k:

The set of equations to solve for λ_k are obtained using $\partial\mathscr{L}/\partial\mathbf{q}_k = 0$. It follows that

$$\begin{aligned}0 = h\left[\frac{\beta_0}{h^2}\frac{\partial \mathbf{R}_k}{\partial \ddot{\mathbf{q}}} + \frac{\alpha_0}{h}\frac{\partial \mathbf{R}_k}{\partial \dot{\mathbf{q}}} + \frac{\partial \mathbf{R}_k}{\partial \mathbf{q}}\right]^T \lambda_k + h\sum_{i=1}^{p}\frac{\alpha_i}{h}\frac{\partial \mathbf{R}_{k+i}}{\partial \dot{\mathbf{q}}_{k+i}}^T \lambda_{k+i} + h\sum_{i=1}^{2p}\frac{\beta_i}{h^2}\frac{\partial \mathbf{R}_{k+i}}{\partial \ddot{\mathbf{q}}_{k+i}}^T \lambda_{k+i} \\ + h\left\{\frac{\beta_0}{h^2}\frac{\partial F_k}{\partial \ddot{\mathbf{q}}} + \frac{\alpha_0}{h}\frac{\partial F_k}{\partial \dot{\mathbf{q}}} + \frac{\partial F_k}{\partial \mathbf{q}}\right\}^T + h\sum_{i=1}^{p}\frac{\alpha_i}{h}\frac{\partial F_{k+i}}{\partial \dot{\mathbf{q}}_{k+i}}^T + h\sum_{i=1}^{2p}\frac{\beta_i}{h^2}\frac{\partial F_{k+i}}{\partial \ddot{\mathbf{q}}_{k+i}}^T \\ + \sum_{i=0}^{p}\frac{\alpha_i}{h}\psi_{k+i} + \sum_{i=0}^{2p}\frac{\beta_i}{h^2}\phi_{k+i}\end{aligned} \tag{11}$$

The $9p+8$ terms in Equation (11) arise from the occurrences of the states, $\mathbf{q}_k$, as arguments to equations at different time steps, as illustrated in Figure 2. The contributions due to ψ_k and ϕ_k are zero, which eliminates $3p+2$ terms. Finally, rearranging the terms and dividing by h yields the following linear system to solve for λ_k:

$$\begin{aligned}\left[\frac{\beta_0}{h^2}\frac{\partial \mathbf{R}_k}{\partial \ddot{\mathbf{q}}} + \frac{\alpha_0}{h}\frac{\partial \mathbf{R}_k}{\partial \dot{\mathbf{q}}} + \frac{\partial \mathbf{R}_k}{\partial \mathbf{q}}\right]^T \lambda_k = -\left\{\frac{\beta_0}{h^2}\frac{\partial F_k}{\partial \ddot{\mathbf{q}}} + \frac{\alpha_0}{h}\frac{\partial F_k}{\partial \dot{\mathbf{q}}} + \frac{\partial F_k}{\partial \mathbf{q}}\right\}^T \\ -\sum_{i=1}^{p}\frac{\alpha_i}{h}\frac{\partial \mathbf{R}_{k+i}}{\partial \dot{\mathbf{q}}_{k+i}}^T \lambda_{k+i} - \sum_{i=1}^{2p}\frac{\beta_i}{h^2}\frac{\partial \mathbf{R}_{k+i}}{\partial \ddot{\mathbf{q}}_{k+i}}^T \lambda_{k+i} \\ -\sum_{i=1}^{p}\frac{\alpha_i}{h}\frac{\partial F_{k+i}}{\partial \dot{\mathbf{q}}_{k+i}}^T - \sum_{i=1}^{2p}\frac{\beta_i}{h^2}\frac{\partial F_{k+i}}{\partial \ddot{\mathbf{q}}_{k+i}}^T,\end{aligned} \tag{12}$$

with $6p+6$ terms.

Evaluation of Total Derivative: Once the adjoint variables have been determined, the total derivative of the functional of interest with respect to the design variables, $\mathbf{x}$, is the following linear combination:

$$\frac{df}{d\mathbf{x}} = \sum_{k=0}^{N} h\frac{\partial F_k}{\partial \mathbf{x}} + \sum_{k=0}^{N} h\lambda_k^T\frac{\partial \mathbf{R}_k}{\partial \mathbf{x}} + \sum_{k=0}^{N}\psi_k^T\frac{\partial \mathbf{S}_k}{\partial \mathbf{x}} + \sum_{k=0}^{N}\phi_k^T\frac{\partial \mathbf{T}_k}{\partial \mathbf{x}}. \tag{13}$$

Since, the state approximation equations $\mathbf{S}_k$ and $\mathbf{T}_k$ are independent of the design variables, $\mathbf{x}$, it follows that

$$\frac{df}{d\mathbf{x}} = \sum_{k=0}^{N} h\frac{\partial F_k}{\partial \mathbf{x}} + \sum_{k=0}^{N} h\lambda_k^T\frac{\partial \mathbf{R}_k}{\partial \mathbf{x}}. \tag{14}$$

B. Newmark Method

The Newmark family of integrators are single-step methods that use state variable values and their time derivatives from the previous step. The pioneering work of this method were by Fox-Goodwin [17] and Newmark [34], and was originally developed for the numerical solution of problems in structural dynamics including linear elastic studies, dynamic loading and vibrations due to earthquake. It has subsequently found applications in flexible multibody dynamics over the years. The coefficients of this methods are β and γ, that assume the values listed in Table 2, for different variants in the family.

1. Solution of the State Variables

STATE APPROXIMATIONS: The primary unknowns of the Newmark method are the second time derivatives of the state variables, $\ddot{\mathbf{q}}_k$, at each time step, k. The p-th order state approximations are,

$$\begin{aligned} \dot{\mathbf{q}}_k &= \dot{\mathbf{q}}_{k-1} + (1-\gamma)h\ddot{\mathbf{q}}_{k-1} + \gamma h\ddot{\mathbf{q}}_k + \mathscr{O}(h^p) \\ \mathbf{q}_k &= \mathbf{q}_{k-1} + h\dot{\mathbf{q}}_{k-1} + \frac{1-2\beta}{2}h^2\ddot{\mathbf{q}}_{k-1} + \beta h^2\ddot{\mathbf{q}}_k + \mathscr{O}(h^p) \end{aligned} \tag{15}$$

The state approximations (15) are simply weighted linear combinations of vectors as illustrated in Figure 3. The order of accuracy, p, of the Newmark scheme depends on the choice of the coefficients, β and γ, as shown in Table 2.

$\mathbf{S}_k$
$\dot{q}_{k-1}$
$\ddot{q}_{k-1}$
$(1-\gamma)h$
1
$\ddot{q}_k$
γh
$\dot{q}_k$
q_{k-1}
$\dot{q}_{k-1}$
$\mathbf{T}_k$
1
h
$\frac{1-2\beta}{2}h^2$
$\ddot{q}_{k-1}$
q_k
βh^2
$\ddot{q}_k$

Figure 3: An illustration of weighted linear combination of states yielding the velocity (left) and position (right) states for Newmark method. These constitute the state approximation equations $\mathbf{S}_k$ and $\mathbf{T}_k$.

Table 2: The coefficients of Newmark family of methods and their corresponding orders of accuracy.

Method	β	γ	Order
Implicit Fox-Goodwin [17]	1/12	1/2	3
Implicit linear acceleration	1/6	1/2	2
Implicit average constant acceleration	1/4	1/2	2
Implicit central difference	0	1/2	2
Explicit	0	0	1

SOLUTION OF THE NONLINEAR SYSTEM: The nonlinear governing equations at each step, $\mathbf{R}_k(\ddot{\mathbf{q}}_k, \dot{\mathbf{q}}_k, \mathbf{q}_k, \mathbf{x}, t_k) = 0$, are linearized with respect to $\ddot{\mathbf{q}}_k$ as follows

$$\left[\frac{\partial \mathbf{R}_k}{\partial \ddot{\mathbf{q}}} + \gamma h \frac{\partial \mathbf{R}_k}{\partial \dot{\mathbf{q}}} + \beta h^2 \frac{\partial \mathbf{R}_k}{\partial \mathbf{q}}\right] \Delta\ddot{\mathbf{q}}_k = -\mathbf{R}_k(\ddot{\mathbf{q}}_k, \dot{\mathbf{q}}_k, \mathbf{q}_k, \mathbf{x}, t_k). \tag{16}$$

The state variables and its first time derivatives (velocities) are approximated using Equation (15), for an estimated value of the acceleration state variables (second time derivatives), at a known time, t_k. The linear system (16) is then solved for the primary update $\Delta\ddot{\mathbf{q}}_k^n$ (incremental accelerations), at each iteration, n, of the nonlinear solution. The secondary and tertiary updates required for the state variables and their first time derivatives are readily obtained by scaling the acceleration update using the Newmark coefficients. The resulting update formulas are

$$\begin{aligned}
\ddot{\mathbf{q}}_k^{n+1} &= \ddot{\mathbf{q}}_k^{n} + \Delta\ddot{\mathbf{q}}_k^{n}, \\
\dot{\mathbf{q}}_k^{n+1} &= \dot{\mathbf{q}}_k^{n} + \gamma h \Delta\ddot{\mathbf{q}}_k^{n}, \\
\mathbf{q}_k^{n+1} &= \mathbf{q}_k^{n} + \beta h^2 \Delta\ddot{\mathbf{q}}_k^{n}.
\end{aligned} \tag{17}$$

The iterative updates to the state variables and their derivatives are continued until the governing equations are solved to the required tolerance.

2. *Solution of the Adjoint Variables*

FORMATION OF THE LAGRANGIAN: The governing equations and functional of interest follow the same treatment discussed for the BDF method. The state approximations of the Newmark time marching scheme given in Equation (15) are reformulated as following residuals

$$\begin{aligned}
\mathbf{S}_k &= \dot{\mathbf{q}}_{k-1} + (1-\gamma)h\ddot{\mathbf{q}}_{k-1} + \gamma h \ddot{\mathbf{q}}_k - \dot{\mathbf{q}}_k, \\
\mathbf{T}_k &= \mathbf{q}_{k-1} + h\dot{\mathbf{q}}_{k-1} + \frac{1-2\beta}{2} h^2 \ddot{\mathbf{q}}_{k-1} + \beta h^2 \ddot{\mathbf{q}}_k - \mathbf{q}_k.
\end{aligned} \tag{18}$$

The adjoint variables, λ_k, ψ_k and ϕ_k are introduced as respective unknown weights, to the governing equations, $\mathbf{R}_k$, the state approximation equations, $\mathbf{S}_k$, and $\mathbf{T}_k$, arising from the Newmark scheme, for each time step, k. The Lagrangian is formed as the following linear combination:

$$\mathscr{L} = \sum_{k=0}^{N} hF_k + \sum_{k=0}^{N} h\lambda_k^T \mathbf{R}_k + \sum_{k=0}^{N} \psi_k^T \mathbf{S}_k + \sum_{k=0}^{N} \phi_k^T \mathbf{T}_k. \tag{19}$$

The central idea is to represent the functional, F_k, as a linear combination of the other equations. An illustration of the formation of Lagrangian from different equations is shown in Figure 4.

THE ADJOINT EQUATIONS: The system of equations to solve for the adjoint variables is obtained from the stationary points of the Lagrangian with respect the state variables and their time derivatives at each time step. The number of partial derivative terms that exist in the adjoint system of equations, can be graphically determined from Figure 4, based on the occurrences of the corresponding primal variables as inputs to residuals, and is listed in Table 3.

Table 3: The number of terms in the adjoint system of equations for Newmark family of integrators.

	T	**S**	**R**	F	Total
ϕ_k	2		1	1	4
ψ_k	1	2	2	2	7
λ_k	2	2	5	5	14

1. Equation for ϕ_k:

Setting $\partial\mathscr{L}/\partial\mathbf{q}_k = 0$ yields

$$\frac{\partial \mathbf{T}_k}{\partial \mathbf{q}_k}^T \phi_k + \frac{\partial \mathbf{T}_{k+1}}{\partial \mathbf{q}_k}^T \phi_{k+1} + h\frac{\partial \mathbf{R}_{k+1}}{\partial \mathbf{q}_k}^T \lambda_{k+1} + h\frac{\partial F_{k+1}}{\partial \mathbf{q}_k}^T = 0. \tag{20}$$

Further simplifications result in

$$\phi_k = \phi_{k+1} + h\left[\frac{\partial \mathbf{R}_{k+1}}{\partial \mathbf{q}_{k+1}}\right]^T \lambda_{k+1} + h\left\{\frac{\partial F_{k+1}}{\partial \mathbf{q}_{k+1}}\right\}^T. \tag{21}$$

It can be noticed that the four terms of Equation (21) correspond to the occurrences of the primal variable $\mathbf{q}_k$ as inputs to equations during forward solution history.

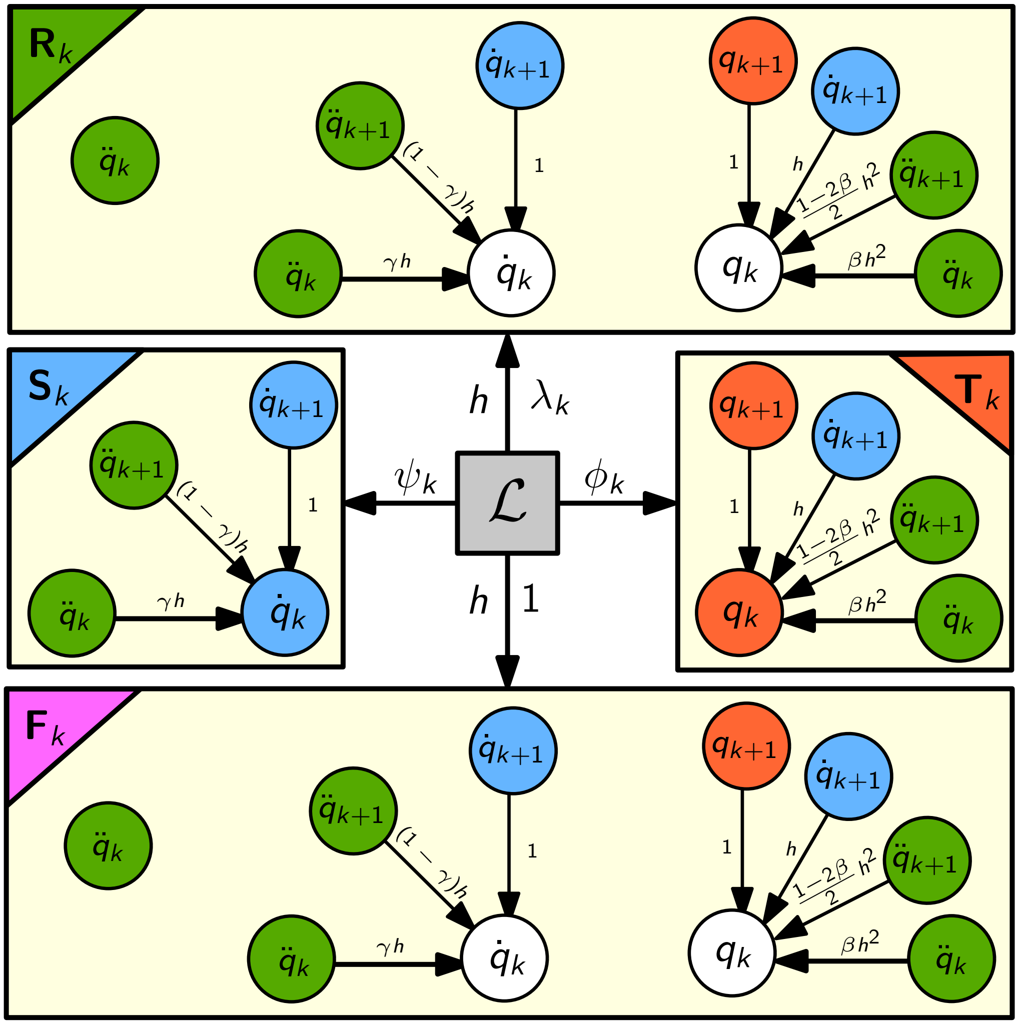


Figure 4: The weighted linear combinations of *equations* with corresponding *adjoint variables* forming the Lagrangian for Newmark method.

2. Equation for ψ_k:

Using $\partial\mathscr{L}/\partial\dot{\mathbf{q}}_k = 0$:

$$\frac{\partial \mathbf{S}_k}{\partial \dot{\mathbf{q}}_k}^T \psi_k + \frac{\partial \mathbf{S}_{k+1}}{\partial \dot{\mathbf{q}}_k}^T \psi_{k+1} + \frac{\partial \mathbf{T}_{k+1}}{\partial \dot{\mathbf{q}}_k}^T \phi_{k+1} + h\frac{\partial \mathbf{R}_{k+1}}{\partial \dot{\mathbf{q}}_k}^T \lambda_{k+1} + h\frac{\partial F_{k+1}}{\partial \dot{\mathbf{q}}_k}^T = 0. \tag{22}$$

This simplifies to

$$\psi_k = \psi_{k+1} + h\phi_{k+1} + h\left[\frac{\partial \mathbf{R}_{k+1}}{\partial \dot{\mathbf{q}}_{k+1}} + h\frac{\partial \mathbf{R}_{k+1}}{\partial \mathbf{q}_{k+1}}\right]^T \lambda_{k+1} + h\left\{\frac{\partial F_{k+1}}{\partial \dot{\mathbf{q}}_{k+1}} + h\frac{\partial F_{k+1}}{\partial \mathbf{q}_{k+1}}\right\}^T. \tag{23}$$

The seven terms in Equation (23) represent and accumulate contributions from residuals/equations that were affected by $\dot{\mathbf{q}}_k$ during the forward mode.

3. Equation for λ_k:

Setting $\partial\mathscr{L}/\partial\ddot{\mathbf{q}}_k = 0$ yields the equation to solve for the adjoint variable, λ_k,

$$h\sum_{i=0}^{1}\frac{\partial \mathbf{R}_{k+i}}{\partial \ddot{\mathbf{q}}_k}^T \lambda_{k+i} + h\sum_{i=0}^{1}\frac{\partial F_{k+i}}{\partial \ddot{\mathbf{q}}_k}^T + \sum_{i=0}^{1}\frac{\partial \mathbf{S}_{k+i}}{\partial \ddot{\mathbf{q}}_k}^T \psi_{k+i} + \sum_{i=0}^{1}\frac{\partial \mathbf{T}_{k+i}}{\partial \ddot{\mathbf{q}}_k}^T \phi_{k+i} = 0. \tag{24}$$

Rearranging the terms results in the following linear system for λ_k

$$\begin{aligned}\left[\frac{\partial \mathbf{R}_k}{\partial \ddot{\mathbf{q}}_k}+\gamma h\frac{\partial \mathbf{R}_k}{\partial \dot{\mathbf{q}}_k}+\beta h^2\frac{\partial \mathbf{R}_k}{\partial \mathbf{q}_k}\right]^T\lambda_k = &-\left\{\frac{\partial F_k}{\partial \ddot{\mathbf{q}}_k}+\gamma h\frac{\partial F_k}{\partial \dot{\mathbf{q}}_k}+\beta h^2\frac{\partial F_k}{\partial \mathbf{q}_k}\right\}^T\\ &-\frac{1}{h}\left\{\gamma h\psi_k+\beta h^2\phi_k\right\}^T\\ &-\left[(1-\gamma)h\frac{\partial \mathbf{R}_{k+1}}{\partial \dot{\mathbf{q}}_{k+1}}+\frac{1-2\beta}{2}h^2\frac{\partial \mathbf{R}_{k+1}}{\partial \mathbf{q}_{k+1}}\right]^T\lambda_{k+1}\\ &-\left\{(1-\gamma)h\frac{\partial F_{k+1}}{\partial \dot{\mathbf{q}}_{k+1}}+\frac{1-2\beta}{2}h^2\frac{\partial F_{k+1}}{\partial \mathbf{q}_{k+1}}\right\}^T\\ &-\frac{1}{h}\left\{(1-\gamma)h\psi_{k+1}+\frac{1-2\beta}{2}h^2\phi_{k+1}\right\}^T.\end{aligned}\tag{25}$$

The fourteen terms can be geometrically interpreted in Figure 4, as occurrences of the primal state variable $\ddot{\mathbf{q}}_k$ as inputs argument to equations in the time history. The determination of adjoint variables allows evaluating the total derivative of the functional of interest with respect to the design variables, using Equation (14).

C. Adams–Bashforth–Moulton

The Adams–Bashforth–Moulton [5, 32] family of linear multistep methods use the past solution values to construct the solution at current step. The ABM methods are based on numerical integration of the polynomial that interpolates solution values. The number of values used to construct the solution determines the order of accuracy and stability of the method. The ABM method is a part of packages such as `EPISODE` [11] and `LSODE` [24], and has been applied to solve stiff problems. The ABM method is the popular alternative to Runge–Kutta method when the evaluation of governing equations are very expensive.

1. *Solution of the State Variables*

The primary unknowns of the ABM family of methods [34] are the acceleration state variables, $\ddot{\mathbf{q}}_k$, at each time step, k. The first time derivative of state variables, $\dot{\mathbf{q}}_k$, are obtained by numerical integration of the second time derivative of states as follows

$$\dot{\mathbf{q}}_k = \dot{\mathbf{q}}_{k-1}+\sum_{i=0}^{p-1} h\alpha_i\ddot{\mathbf{q}}_{k-i}+\mathscr{O}(h^p).\tag{26}$$

The state variables, $\mathbf{q}_k$, are obtained by numerical integration of the first time derivative of states as follows

$$\mathbf{q}_k = \mathbf{q}_{k-1}+\sum_{i=0}^{p-1} h\alpha_i\dot{\mathbf{q}}_{k-i}+\mathscr{O}(h^p).\tag{27}$$

The schematic representation of the ABM state approximations is shown in Figure 5.

SOLUTION OF THE NONLINEAR SYSTEM: The system of governing equations, $\mathbf{R}_k(\ddot{\mathbf{q}}_k,\dot{\mathbf{q}}_k,\mathbf{q}_k,\mathbf{x},t_k)=0$, is linearized with respect to the primary unknown variable, $\ddot{\mathbf{q}}_k$, of the ABM time marching scheme at each time step, k, as follows

$$\left[\frac{\partial \mathbf{R}_k}{\partial \ddot{\mathbf{q}}}+h\alpha_0\frac{\partial \mathbf{R}_k}{\partial \dot{\mathbf{q}}}+h^2\alpha_0^2\frac{\partial \mathbf{R}_k}{\partial \mathbf{q}}\right]\Delta\ddot{\mathbf{q}}_k = -\mathbf{R}_k(\ddot{\mathbf{q}}_k,\dot{\mathbf{q}}_k,\mathbf{q}_k,\mathbf{x},t_k).\tag{28}$$

The state variables and the first time derivatives are approximated using Equations (27) and (26), for an estimated value of $\ddot{\mathbf{q}}_k$. The linearization of the governing equations (28) is then solved for the primary update $\Delta\ddot{\mathbf{q}}_k^n$, at each iteration, n, of the nonlinear solution. The secondary updates to state variables are readily obtained by scaling the primary update using the ABM coefficients. The resulting update formulas to the state variables and their time derivatives are

$$\begin{aligned}\ddot{\mathbf{q}}_k^{n+1} &= \ddot{\mathbf{q}}_k^n+\Delta\ddot{\mathbf{q}}_k^n,\\ \dot{\mathbf{q}}_k^{n+1} &= \dot{\mathbf{q}}_k^n+h\alpha_0\Delta\ddot{\mathbf{q}}_k^n,\\ \mathbf{q}_k^{n+1} &= \mathbf{q}_k^n+h^2\alpha_0^2\Delta\ddot{\mathbf{q}}_k^n.\end{aligned}\tag{29}$$

The iterative updates to the state variables and their time derivatives are continued until the governing equations are solved to required tolerance.

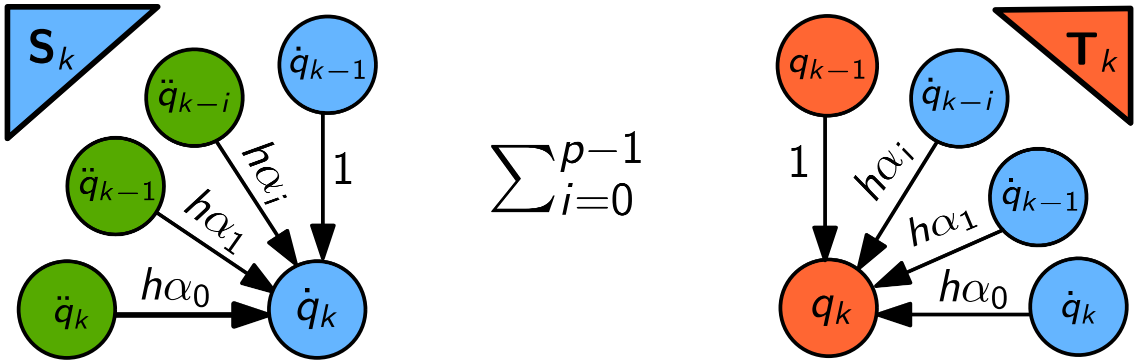


Figure 5: A weighted linear combination of state variables with scaled ABM coefficients yielding the first time derivative of states (left) and the state variables (right).

2. *Solution of the Adjoint Variables*

FORMATION OF THE LAGRANGIAN: The governing equations and functional of interest follow same treatment discussed previously for other methods. The state approximations of the ABM time marching scheme given in equations (26) and (27) are expressed as the following residuals

$$\begin{aligned}
\mathbf{S}_k &= \dot{\mathbf{q}}_{k-1} + h\sum_{i=0}^{p-1}\alpha_i\ddot{\mathbf{q}}_{k-i} - \dot{\mathbf{q}}_{\mathbf{k}}, \\
\mathbf{T}_k &= \mathbf{q}_{k-1} + h\alpha_0\left(\dot{\mathbf{q}}_{k-1} + \sum_{i=0}^{p-1} h\alpha_i\ddot{\mathbf{q}}_{k-i}\right) + \sum_{i=1}^{p-1} h\alpha_i\dot{\mathbf{q}}_{k-i} - \mathbf{q}_k.
\end{aligned} \tag{30}$$

The term $h\alpha_0\dot{\mathbf{q}}_k$ is expanded out in terms of the primary unknown $\ddot{\mathbf{q}}_k$ for eliminating the coupling of adjoint equations within each time step. The adjoint variables, λ_k, ψ_k and ϕ_k are introduced as respective unknown weights, to the governing equations, $\mathbf{R}_k$, the state approximation equations, $\mathbf{S}_k$, and $\mathbf{T}_k$, arising from the ABM scheme, for each time step, k. The Lagrangian is written as

$$\mathscr{L} = \sum_{k=0}^{N} hF_k + \sum_{k=0}^{N} h\lambda_k^T\mathbf{R}_k + \sum_{k=0}^{N}\psi_k^T\mathbf{S}_k + \sum_{k=0}^{N}\phi_k^T\mathbf{T}_k. \tag{31}$$

The formation of the Lagrangian for the ABM method is illustrated in Figure 6. The number of terms that occur in the adjoint system of equations for each adjoint variable can be determined graphically from Figure 6 and is listed in Table 4.

THE ADJOINT EQUATIONS: The system of equations to solve for the adjoint variables is obtained from the stationary points of the Lagrangian with respect to the position, velocity and acceleration state variables, at each time step, k.

Table 4: Table listing the number of terms in the adjoint system of equations for ABM method.

	T	**S**	**R**	F	Total
ϕ_k	2		1	1	4
ψ_k	p	2	p+1	p+1	3p+4
λ_k	p	p	2p+1	2p+1	6p+2

1. Equation for ϕ_k:

Setting $\partial\mathscr{L}/\partial\mathbf{q}_k = 0$ yields

$$\frac{\partial\mathbf{T}_k}{\partial\mathbf{q}_k}^T\phi_k + \frac{\partial\mathbf{T}_{k+1}}{\partial\mathbf{q}_k}^T\phi_{k+1} + h\frac{\partial\mathbf{R}_{k+1}}{\partial\mathbf{q}_k}^T\lambda_{k+1} + h\frac{\partial F_{k+1}}{\partial\mathbf{q}_k}^T = 0. \tag{32}$$

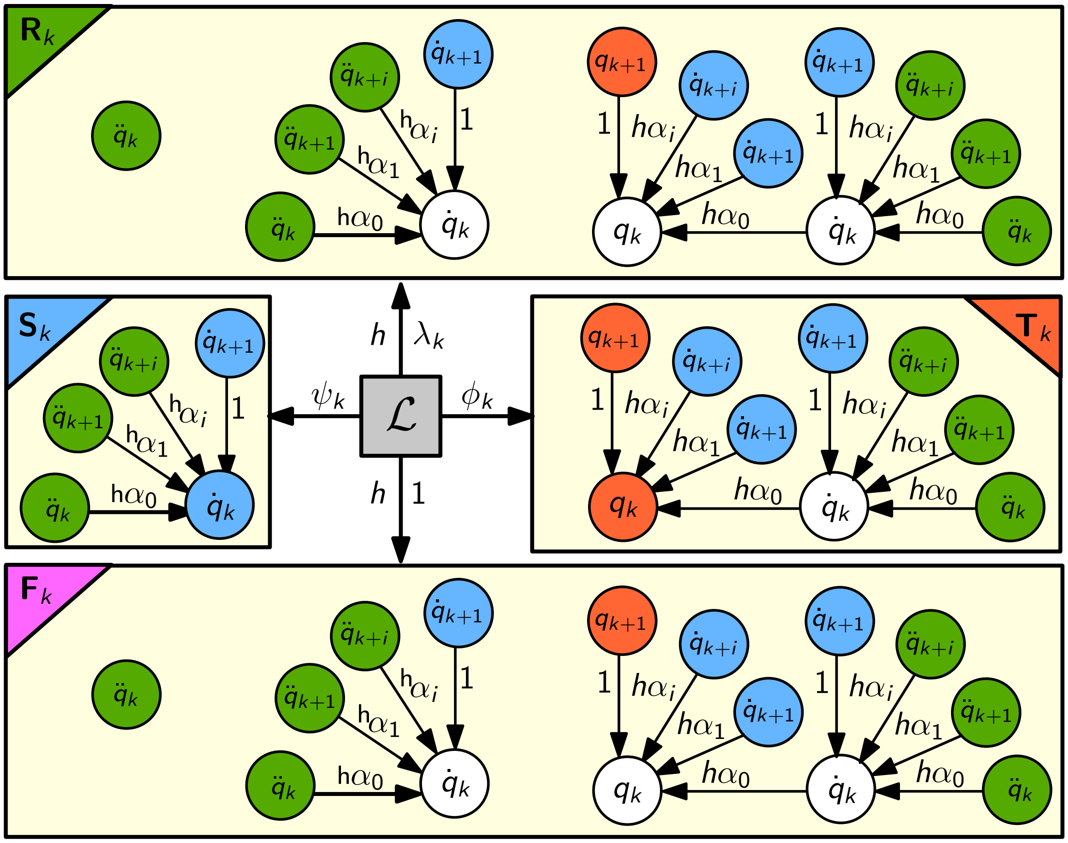


Figure 6: A weighted linear combination of *equations* with corresponding *adjoint variables* forming the Lagrangian for ABM method.

This simplifies to

$$\phi_k = \phi_{k+1} + h\left[\frac{\partial \mathbf{R}_{k+1}}{\partial \mathbf{q}_{k+1}}\right]^T \lambda_{k+1} + h\left\{\frac{\partial F_{k+1}}{\partial \mathbf{q}_{k+1}}\right\}^T. \tag{33}$$

2. Equation for ψ_k:

Setting $\partial \mathscr{L}/\partial \dot{\mathbf{q}}_k = 0$ yields

$$\frac{\partial \mathbf{S}_k}{\partial \dot{\mathbf{q}}_k}^T \psi_k + \frac{\partial \mathbf{S}_{k+1}}{\partial \dot{\mathbf{q}}_k}^T \psi_{k+1} + \sum_{i=1}^{p-1} \frac{\partial \mathbf{T}_{k+i}}{\partial \dot{\mathbf{q}}_k}^T \phi_{k+i} + h\sum_{i=1}^{p-1} \frac{\partial \mathbf{R}_{k+i}}{\partial \dot{\mathbf{q}}_k}^T \lambda_{k+i} + h\sum_{i=1}^{p-1} \frac{\partial F_{k+i}}{\partial \dot{\mathbf{q}}_k}^T = 0. \tag{34}$$

Further simplifications result in the adjoint variable, ψ_k, as a linear combination:

$$\begin{aligned}\psi_k = \psi_{k+1} + h\alpha_0\phi_{k+1} + h\left[\frac{\partial \mathbf{R}_{k+1}}{\partial \dot{\mathbf{q}}_{k+1}} + h\alpha_0 \frac{\partial \mathbf{R}_{k+1}}{\partial \mathbf{q}_{k+1}}\right]^T \lambda_{k+1} + h\left\{\frac{\partial F_{k+1}}{\partial \dot{\mathbf{q}}_{k+1}} + h\alpha_0 \frac{\partial F_{k+1}}{\partial \mathbf{q}_{k+1}}\right\}^T \\ + h\sum_{i=1}^{p-1} \alpha_i \phi_{k+i} + h\sum_{i=1}^{p-1}\left[h\alpha_i \frac{\partial \mathbf{R}_{k+i}}{\partial \mathbf{q}_{k+i}}\right]^T \lambda_{k+i} + h\sum_{i=1}^{p-1}\left\{h\alpha_i \frac{\partial F_{k+i}}{\partial \mathbf{q}_{k+i}}\right\}^T.\end{aligned} \tag{35}$$

3. Equation for λ_k:

Setting $\partial \mathscr{L}/\partial \ddot{\mathbf{q}}_k = 0$ yields the equation to solve for the adjoint variable, λ_k,

$$h\sum_{i=0}^{p-1} \frac{\partial \mathbf{R}_{k+i}}{\partial \ddot{\mathbf{q}}_k}^T \lambda_{k+i} + h\sum_{i=0}^{p-1} \frac{\partial F_{k+i}}{\partial \ddot{\mathbf{q}}_k}^T + \sum_{i=0}^{p-1} \frac{\partial \mathbf{S}_{k+i}}{\partial \ddot{\mathbf{q}}_k}^T \psi_{k+i} + \sum_{i=0}^{p-1} \frac{\partial \mathbf{T}_{k+i}}{\partial \ddot{\mathbf{q}}_k}^T \phi_{k+i} = 0. \tag{36}$$

Expanding the derivative terms, separating out current and previous determined terms, dividing by h and rearranging for the unknown adjoint variable λ_k results in the following linear system:

$$
\begin{aligned}
\left[\frac{\partial \mathbf{R}_k}{\partial \ddot{\mathbf{q}}_k} + h\alpha_0 \frac{\partial \mathbf{R}_k}{\partial \dot{\mathbf{q}}_k} + h^2\alpha_0^2 \frac{\partial \mathbf{R}_k}{\partial \mathbf{q}_k}\right]^T \lambda_k = & -\left\{\frac{\partial F_k}{\partial \ddot{\mathbf{q}}_k} + h\alpha_0 \frac{\partial F_k}{\partial \dot{\mathbf{q}}_k} + h^2\alpha_0^2 \frac{\partial F_k}{\partial \mathbf{q}_k}\right\}^T \\
& - \frac{1}{h}\left\{h\alpha_0 \psi_k + h^2\alpha_0^2 \phi_k\right\} \\
& - \sum_{i=1}^{p-1} \left[h\alpha_i \frac{\partial \mathbf{R}_{k+i}}{\partial \dot{\mathbf{q}}_{k+i}} + h\alpha_0 h\alpha_i \frac{\partial \mathbf{R}_{k+i}}{\partial \mathbf{q}_{k+i}}\right]^T \lambda_{k+i} \\
& - \sum_{i=1}^{p-1} \left\{h\alpha_i \frac{\partial F_{k+i}}{\partial \dot{\mathbf{q}}_{k+i}} + h\alpha_0 h\alpha_i \frac{\partial F_{k+i}}{\partial \mathbf{q}_{k+i}}\right\}^T \\
& - \frac{1}{h}\sum_{i=1}^{p-1} \left\{h\alpha_i \psi_{k+i} + h\alpha_0 h\alpha_i \phi_{k+i}\right\}.
\end{aligned}
\tag{37}
$$

The partial derivative terms in the linear system correspond to the occurrences of $\ddot{\mathbf{q}}_k$ as inputs to equations $\mathbf{R}$, $\mathbf{S}$, $\mathbf{T}$ and F in the time history, as illustrated in Figure 6.

D. Diagonally Implicit Runge–Kutta

The original Runge–Kutta formulas are explicit and not well-suited for the stiff systems that arise in flexible multibody dynamic simulations. Butcher [10] extended Runge–Kutta methods to include a class of Implicit Runge–Kutta (IRK) methods that can be used to solve stiff problems. As a further refinement, Alexander [3] and subsequently Cash [13] developed Diagonally Implicit Runge–Kutta (DIRK) methods that achieve computational savings compared to IRK methods. The computational savings arise from the lower triangular structure of the matrix of DIRK coefficients. The development of DIRK scheme for second-order descriptor systems and the corresponding time-dependent discrete-adjoint are discussed in the remainder of this section.

1. Solution of the State Variables

BUTCHER'S TABLEAU Runge–Kutta methods belong to the class of multistage methods for solving differential equations. The coefficients that define the Runge–Kutta method can be conveniently arranged in a tabular format known as Butcher's tableau. The structure of the Butcher's tableau for diagonally implicit Runge–Kutta is shown in Table 5. The lower triangular nature of the tableau enables the successive solution of the nonlinear governing equations at each stage. Note that the tableau is complete in the case of an IRK scheme, resulting in full coupling among all stages. Due to the availability of *one-stage-second-order*, *two-stage-third-order* and *three-stage-fourth-order* DIRK methods developed by Alexander [3], the DIRK method is preferred to IRK methods.

Table 5: Butcher's tableau structure for diagonally implicit Runge–Kutta methods.

Stage	β_1	β_2	$\cdots$	β_s	
1	α_{11}	0	0	0	τ_1
2	α_{21}	α_{22}	0	0	τ_2
	$\vdots$	$\vdots$	$\ddots$	0	$\vdots$
s	α_{s1}	α_{s2}	$\cdots$	α_{ss}	τ_s

STAGE APPROXIMATIONS: The governing equations are solved at intermediate time steps, t_{ki}, referred to as the *stages*. The intermediate stage state variables and their first and second time derivatives are denoted as $\mathbf{u}_{ki}$, $\dot{\mathbf{u}}_{ki}$ and $\ddot{\mathbf{u}}_{ki}$, respectively. The stage state approximation relations are

$$
\begin{aligned}
\dot{\mathbf{u}}_{ki} &= \dot{\mathbf{q}}_{k-1} + h\sum_{j=1}^{i} \alpha_{ij}\ddot{\mathbf{u}}_{kj}, \\
\mathbf{u}_{ki} &= \mathbf{q}_{k-1} + h\sum_{j=1}^{i} \alpha_{ij}\dot{\mathbf{u}}_{kj}.
\end{aligned}
\tag{38}
$$

The indices i and j to refer to row and column of the coefficients in the Butcher's tableau shown in Table 5. The schematic representation of the intermediate stage approximations is shown in Figure 7.

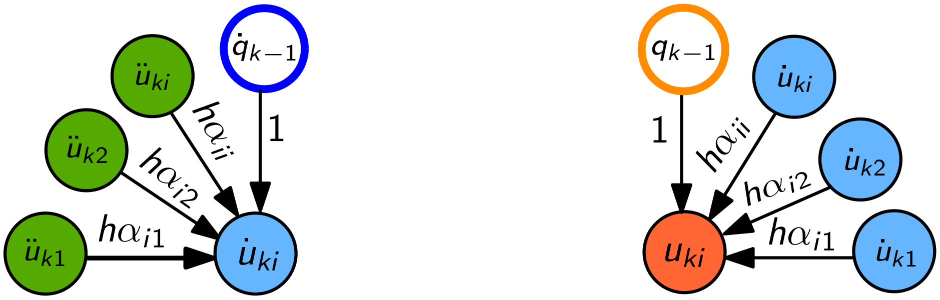


Figure 7: The intermediate stage state variables of DIRK are formed as a linear combination.

State Approximations: The state variables at time, t_k, are expressed as follows

$$\begin{aligned} \ddot{\mathbf{q}}_k &= \sum_{i=1}^{s} \beta_i \ddot{\mathbf{u}}_{ki}, \\ \dot{\mathbf{q}}_k &= \dot{\mathbf{q}}_{k-1} + h\sum_{i=1}^{s} \beta_i \ddot{\mathbf{u}}_{ki}, \\ \mathbf{q}_k &= \mathbf{q}_{k-1} + h\sum_{i=1}^{s} \beta_i \dot{\mathbf{u}}_{ki} \end{aligned} \tag{39}$$

and is illustrated in Figure 8.

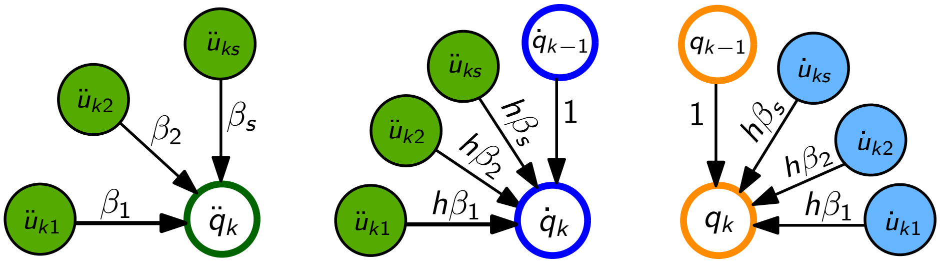


Figure 8: The state variables and their time derivatives at k-th time step formed as a linear combination.

Solution of the Nonlinear System: The nonlinear system of equations to be solved, at each stage, i, and time step, k, is $\mathbf{R}_{ki}(\ddot{\mathbf{u}}_{ki}, \dot{\mathbf{u}}_{ki}, \mathbf{u}_{ki}, \mathbf{x}, t_{ki}) = 0$. This can be solved in a manner similar to Newmark and ABM methods discussed previously in this section. The key difference here is that the nonlinear system is solved at s-intermediate stages for each time step. The linearized form of the nonlinear system to be solved repeatedly at each stage is,

$$\left[\frac{\partial \mathbf{R}_{ki}}{\partial \ddot{\mathbf{u}}} + h\alpha_{ii}\frac{\partial \mathbf{R}_{ki}}{\partial \dot{\mathbf{u}}} + h^2\alpha_{ii}^2\frac{\partial \mathbf{R}_{ki}}{\partial \mathbf{u}}\right]\Delta\ddot{\mathbf{u}}_{ki} = -\mathbf{R}_{ki}(\ddot{\mathbf{u}}_{ki}, \dot{\mathbf{u}}_{ki}, \mathbf{u}_{ki}, \mathbf{x}, t_{ki}). \tag{40}$$

The update formulas for the stage state variables are

$$\begin{aligned} \ddot{\mathbf{u}}_{ki}^{n+1} &= \ddot{\mathbf{u}}_{ki}^{n} + \Delta\ddot{\mathbf{u}}_{ki}^{n}, \\ \dot{\mathbf{u}}_{ki}^{n+1} &= \dot{\mathbf{u}}_{ki}^{n} + h\alpha_{ii}\Delta\ddot{\mathbf{u}}_{ki}^{n}, \\ \mathbf{u}_{ki}^{n+1} &= \mathbf{u}_{ki}^{n} + h^2\alpha_{ii}^2\Delta\ddot{\mathbf{u}}_{ki}^{n}. \end{aligned} \tag{41}$$

The iterative updates to the stage state variables and derivatives continue until the governing equations are solved to the required tolerance.

2. *Solution of the Adjoint Variables*

FORMATION OF THE LAGRANGIAN: The governing equations and functionals of interest follow slightly different treatment for DIRK since it is a multistage method. The state approximations of DIRK time marching scheme given in Equation (39) are reformulated as the following residuals

$$\begin{aligned} \mathbf{S}_k &= \dot{\mathbf{q}}_{k-1} + h\sum_{i=1}^{s}\beta_i\ddot{\mathbf{u}}_{ki} - \dot{\mathbf{q}}_k, \\ \mathbf{T}_k &= \mathbf{q}_{k-1} + h\sum_{i=1}^{s}\beta_i\dot{\mathbf{u}}_{ki} - \mathbf{q}_k. \end{aligned} \tag{42}$$

The adjoint variables λ_{ki}, ψ_k and ϕ_k are associated with the governing equations at each stage, $\mathbf{R}_{ki}$, and the state approximation equations, $\mathbf{S}_k$, and $\mathbf{T}_k$, arising from the DIRK scheme, for each time step, k. The Lagrangian is written as

$$\mathscr{L} = \sum_{k=0}^{N} h\sum_{i=1}^{s}\beta_i F_{ki} + \sum_{k=0}^{N} h\sum_{i=1}^{s}\beta_i\lambda_{ki}^T\mathbf{R}_{ki} + \sum_{k=0}^{N}\psi_k^T\mathbf{S}_k + \sum_{k=0}^{N}\phi_k^T\mathbf{T}_k. \tag{43}$$

Note that h arises from the discretization of the continuous integral into N time intervals and β_i arises from the discretization of each time interval into s stages. The formation of the Lagrangian is schematically shown in Figure 9. The number of terms that occur in the adjoint system of equations for each adjoint variable can be determined graphically from Figure 9 and is listed in Table 6. The number of terms listed in Table 6 can be seen to exist in the following adjoint system of equations.

Table 6: Table listing the number of terms in the adjoint system of equations for DIRK method.

	T	**S**	**R**	F	Total
ϕ_k	2		s	s	2+2s
ψ_k	s	2	2s	2s	2+5s
λ_k	s-i+1	1	2(s-i)+1	2(s-i)+1	5(s-i)+1

1. Equation for ϕ_k:

Setting $\partial\mathscr{L}/\partial\mathbf{q}_k = 0$ yields

$$\frac{\partial\mathbf{T}_k}{\partial\mathbf{q}_k}^T\phi_k + \frac{\partial\mathbf{T}_{k+1}}{\partial\mathbf{q}_k}^T\phi_{k+1} + h\sum_{i=1}^{s}\beta_i\frac{\partial\mathbf{R}_{k+1,i}}{\partial\mathbf{q}_k}^T\lambda_{k+1,i} + h\sum_{i=1}^{s}\beta_i\frac{\partial F_{k+1,i}}{\partial\mathbf{q}_k}^T = 0. \tag{44}$$

This simplifies to

$$\phi_k = \phi_{k+1} + h\sum_{i=1}^{s}\beta_i\frac{\partial\mathbf{R}_{k+1,i}}{\partial\mathbf{u}_{k+1,i}}^T\lambda_{k+1,i} + h\sum_{i=1}^{s}\beta_i\frac{\partial F_{k+1,i}}{\partial\mathbf{u}_{k+1,i}}^T. \tag{45}$$

2. Equation for ψ_k:

Setting $\partial\mathscr{L}/\partial\dot{\mathbf{q}}_k = 0$ yields

$$\frac{\partial\mathbf{S}_k}{\partial\dot{\mathbf{q}}_k}^T\psi_k + \frac{\partial\mathbf{S}_{k+1}}{\partial\dot{\mathbf{q}}_k}^T\psi_{k+1} + \sum_{i=1}^{s}\beta_i\frac{\partial\mathbf{T}_{k+1}}{\partial\dot{\mathbf{q}}_k}^T\phi_{k+1} + h\sum_{i=1}^{s}\beta_i\frac{\partial\mathbf{R}_{k+1,i}}{\partial\dot{\mathbf{q}}_k}^T\lambda_{k+1,i} + h\sum_{i=1}^{s}\beta_i\frac{\partial F_{k+1,i}}{\partial\dot{\mathbf{q}}_k}^T = 0. \tag{46}$$

This becomes

$$\begin{aligned} \psi_k = \psi_{k+1} + h\sum_{i=1}^{s}\beta_i\phi_{k+1} + h\sum_{i=1}^{s}\beta_i\left[\frac{\partial\mathbf{R}_{k+1,i}}{\partial\dot{\mathbf{u}}_{k+1,i}} + h\sum_{j=1}^{i}\alpha_{ij}\frac{\partial\mathbf{R}_{k+1,i}}{\partial\mathbf{u}_{k+1,i}}\right]^T\lambda_{k+1,i} \\ + h\sum_{i=1}^{s}\beta_i\left\{\frac{\partial F_{k+1,i}}{\partial\dot{\mathbf{u}}_{k+1,i}} + h\sum_{j=1}^{i}\alpha_{ij}\frac{\partial F_{k+1,i}}{\partial\mathbf{u}_{k+1,i}}\right\}^T. \end{aligned} \tag{47}$$

Using the properties of DIRK coefficients: $\sum_{i=1}^{s}\beta_i = 1$ and $\sum_{j=1}^{i}\alpha_{ij} = \tau_i$,

$$\psi_k = \psi_{k+1} + h\phi_{k+1} + h\sum_{i=1}^{s}\beta_i\left[\frac{\partial\mathbf{R}_{k+1,i}}{\partial\dot{\mathbf{u}}_{k+1,i}} + h\tau_i\frac{\partial\mathbf{R}_{k+1,i}}{\partial\mathbf{u}_{k+1,i}}\right]^T\lambda_{k+1,i} + h\sum_{i=1}^{s}\beta_i\left\{\frac{\partial F_{k+1,i}}{\partial\dot{\mathbf{u}}_{k+1,i}} + h\tau_i\frac{\partial F_{k+1,i}}{\partial\mathbf{u}_{k+1,i}}\right\}^T. \tag{48}$$

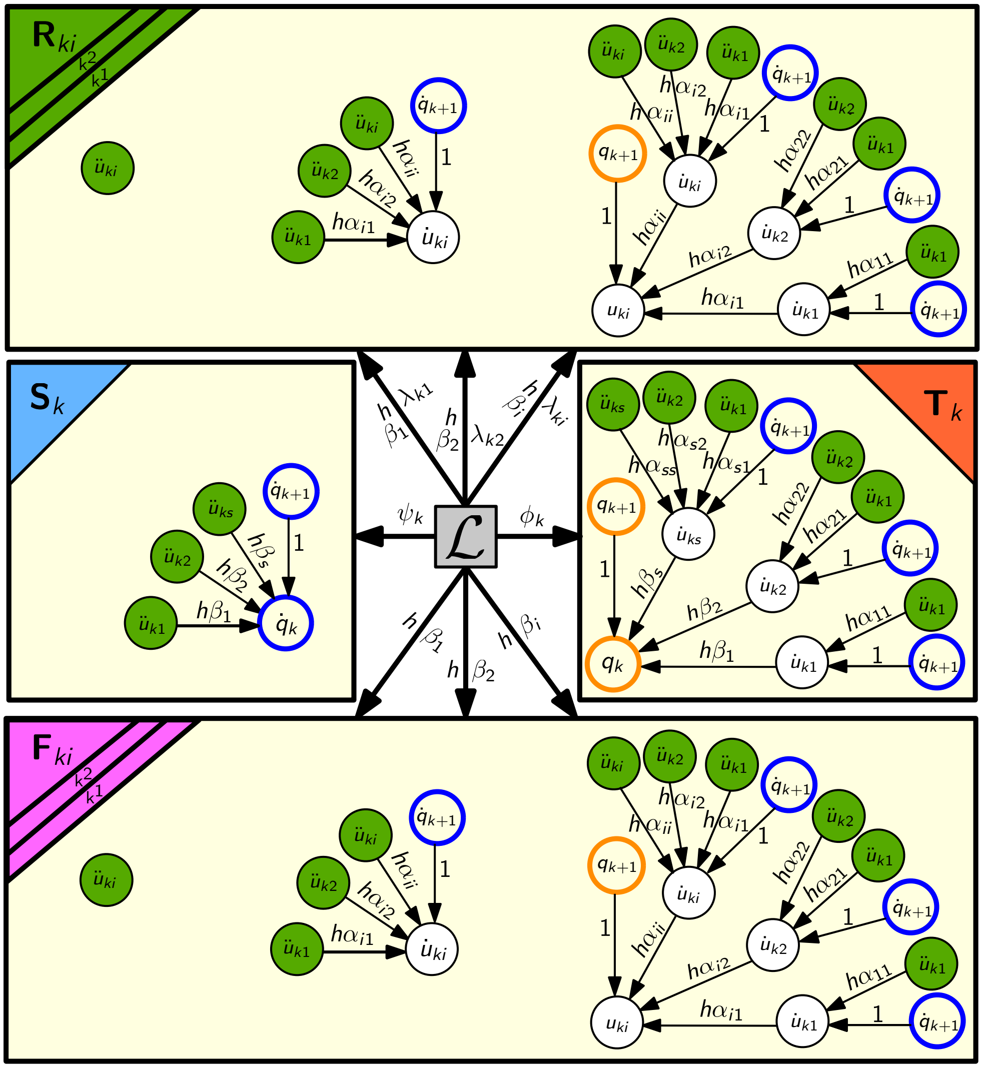


Figure 9: A weighted linear combination of *equations* with corresponding *adjoint variables* forming the Lagrangian for DIRK method.

3. Equation for λ_{ki}:

Taking $\partial \mathscr{L} / \partial \ddot{\mathbf{u}}_{ki} = 0$ yields

$$h\sum_{j=i}^{s} \beta_j \frac{\partial \mathbf{R}_{kj}}{\partial \ddot{\mathbf{u}}_{ki}}^T \lambda_{kj} + h\sum_{j=i}^{s} \beta_j \frac{\partial F_{kj}}{\partial \ddot{\mathbf{u}}_{ki}}^T + \frac{\partial \mathbf{S}_k}{\partial \ddot{\mathbf{u}}_{ki}}^T \psi_k + \frac{\partial \mathbf{T}_k}{\partial \ddot{\mathbf{u}}_{ki}}^T \phi_k = 0. \tag{49}$$

Applying chain rule and expanding the terms results in

$$\begin{aligned} 0 = {} & h\beta_i \frac{\partial \mathbf{R}_{ki}}{\partial \ddot{\mathbf{u}}_{ki}} + h \sum_{j=i+1}^{s} \beta_j \left[h\alpha_{ji} \frac{\partial \mathbf{R}_{kj}}{\partial \dot{\mathbf{u}}_{kj}} + h^2 \sum_{p=i}^{j} \alpha_{jp}\alpha_{pi} \frac{\partial \mathbf{R}_{kj}}{\partial \mathbf{u}_{kj}} \right]^T \lambda_{kj} \\ & + h\beta_i \frac{\partial F_{ki}}{\partial \ddot{\mathbf{u}}_{ki}} + h \sum_{j=i+1}^{s} \beta_j \left\{ h\alpha_{ji} \frac{\partial F_{kj}}{\partial \dot{\mathbf{u}}_{kj}} + h^2 \sum_{p=i}^{j} \alpha_{jp}\alpha_{pi} \frac{\partial F_{kj}}{\partial \mathbf{u}_{kj}} \right\}^T \\ & + h\beta_i \psi_k \quad + \left(h \sum_{j=i}^{s} \beta_j h\alpha_{ji} \right) \phi_k. \end{aligned} \tag{50}$$

Dividing by h and rearranging for the unknown adjoint variable λ_{ki} results in the following linear system

$$
\begin{aligned}
\beta_i \left[\frac{\partial \mathbf{R}_{ki}}{\partial \ddot{\mathbf{u}}_{ki}} + h\alpha_{ii}\frac{\partial \mathbf{R}_{ki}}{\partial \dot{\mathbf{u}}_{ki}} + h^2\alpha_{ii}^2\frac{\partial \mathbf{R}_{ki}}{\partial \mathbf{u}_{ki}}\right]^T \lambda_{ki} = &-\beta_i \left[\frac{\partial F_{ki}}{\partial \ddot{\mathbf{u}}_{ki}} + h^2\alpha_{ii}^2\frac{\partial F_{ki}}{\partial \dot{\mathbf{u}}_{ki}} + h^2\alpha_{ii}^2\frac{\partial F_{ki}}{\partial \mathbf{u}_{ki}}\right]^T \\
&- \sum_{j=i+1}^{s}\beta_j \left[h\alpha_{ji}\frac{\partial \mathbf{R}_{kj}}{\partial \dot{\mathbf{u}}_{kj}} + h^2\sum_{p=i}^{j}\alpha_{jp}\alpha_{pi}\frac{\partial \mathbf{R}_{kj}}{\partial \mathbf{u}_{kj}}\right]^T \lambda_{kj} \\
&- \sum_{j=i+1}^{s}\beta_j \left\{h\alpha_{ji}\frac{\partial F_{kj}}{\partial \dot{\mathbf{u}}_{kj}} + h^2\sum_{p=i}^{j}\alpha_{jp}\alpha_{pi}\frac{\partial F_{kj}}{\partial \mathbf{u}_{kj}}\right\}^T \\
&- \beta_i\psi_k - \left(\sum_{j=i}^{s}\beta_j h\alpha_{ji}\right)\phi_k.
\end{aligned}
\tag{51}
$$

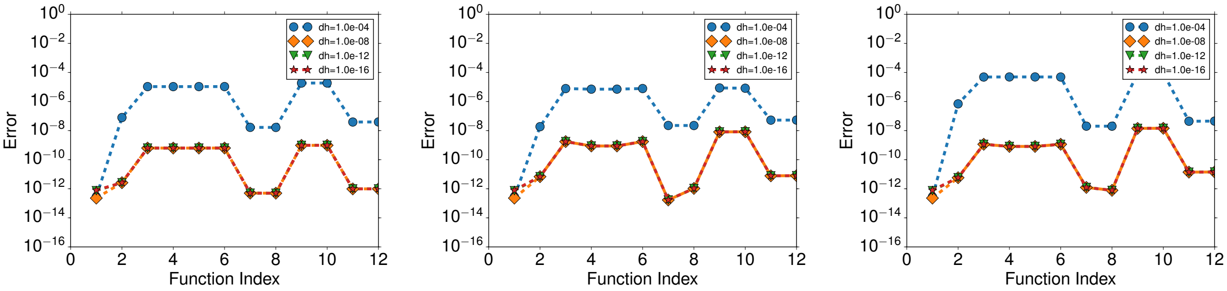
The total derivative is computed in an analogous manner to other methods as follows

$$
\frac{dF}{d\mathbf{x}} = \sum_{k=0}^{N} h\sum_{i=1}^{s}\beta_i\frac{\partial F_{ki}}{\partial \mathbf{x}} + \sum_{k=0}^{N} h\sum_{i=1}^{s}\beta_i\lambda_{ki}^T\frac{\partial \mathbf{R}_{ki}}{\partial \mathbf{x}}. \tag{52}
$$

III. Results

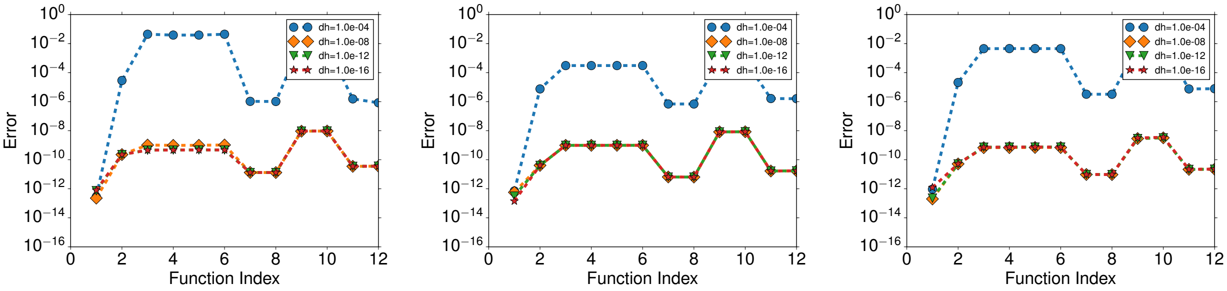
This section presents the verification of the adjoint gradients against the complex-step method, assessments of the parallel scalability of the implementation of the solution procedures, and finally demonstrates applications of the framework on flexible multibody dynamics problems.

A. Numerical Verification of Adjoint Gradients



(a) first-order (b) second-order (c) third-order

Figure 10: Complex-step verification of the BDF adjoint scheme for 12 different functionals with various perturbation step sizes.



(a) second-order (one stage) (b) third-order (two stages) (c) fourth-order (three stages)

Figure 11: Complex-step verification of the DIRK adjoint scheme for 12 different functionals with various perturbation step sizes.

The complex-step method is used to numerically verify the total derivative computed using the adjoint method. The complex-step method does not suffer from subtractive cancellation errors which enables the use of very small step sizes, producing highly accurate gradient estimates. A flexible plate with roughly 1000 degrees of freedom is used as the simulation model for this study. The simulation is run for 1000 time steps using the time marching schemes discussed in Section II, with different orders of accuracy where applicable. The functionals used for this verification are the structural mass (index 1), compliance (index 2), the KS aggregates of the von Mises failure criterion (indices 3 and 4), and the induced exponential aggregates of the von Mises failure criterion (indices 5 to 12). The verification study compares the derivative of these functionals with respect to plate thickness as design variable.

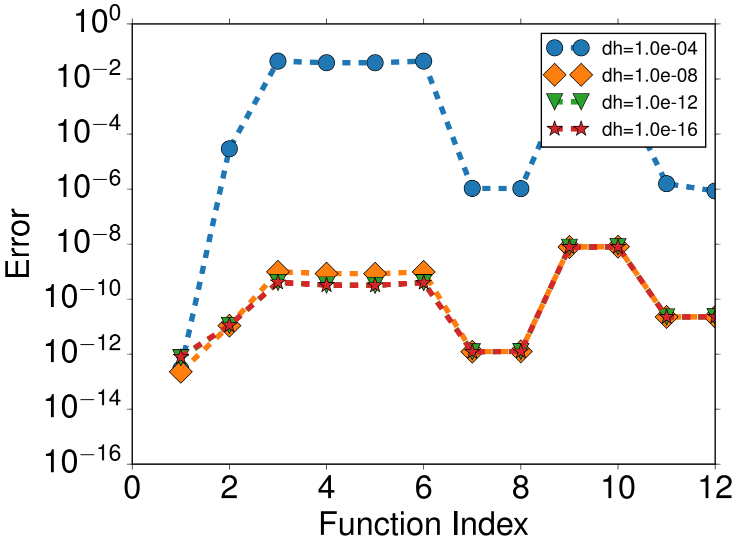
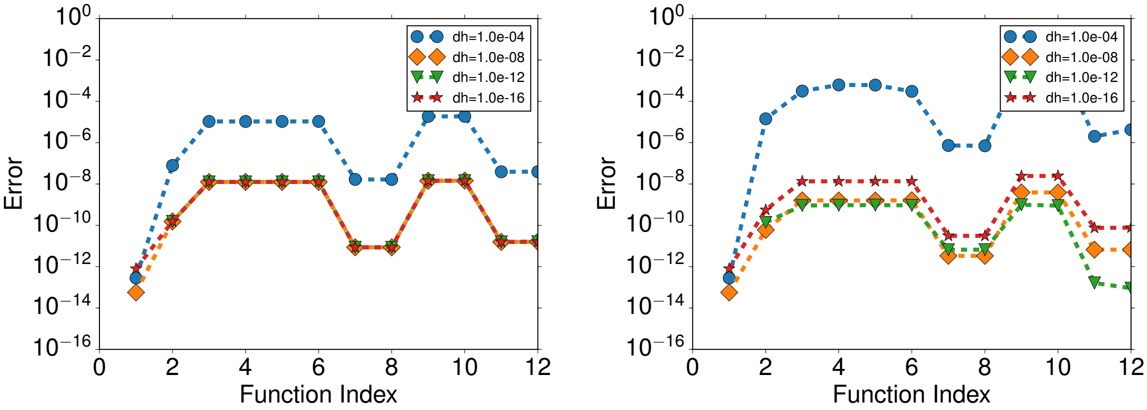


Figure 12: Complex-step verification of the Newmark adjoint scheme for 12 different functionals with various perturbation step sizes.



(a) first-order

(b) second-order

Figure 13: Complex-step verification of the ABM adjoint scheme for 12 different functionals with various perturbation step sizes.

The absolute error between the complex-step derivative and the total derivative computed using the adjoint formulations for the BDF, DIRK, Newmark and ABM time marching schemes are plotted in Figures 10, 11, 12, and 13, respectively. The errors are plotted for perturbation sizes 10^{-4}, 10^{-8}, 10^{-12} and 10^{-12}. The error for the largest perturbation size of 10^{-4}, shown in blue, are predominantly due to truncation, while the error in the all other perturbation sizes are due to round-off errors accumulated over each time step. Overall, each component of the gradient exhibits a relative accuracy better than 10^{-8}.

B. Parallel Scalability Assessments

The parallel scalability of both the time marching method and the adjoint implementation is critical for efficiently solving large problems with several millions of degrees of freedom. In this section, we present a study of the scalability of the proposed framework on a flexible plate model with a medium problem size of 192,000 degrees of freedom and a large problem with 2 million degrees of freedom. The results are organized in two main groups: (1) Scalability with the number of degrees of freedom and (2) Scalability with the number of functionals of interest. We compare both the overall forward and adjoint solution computational performance as well as detailed forward and adjoint operations described below.

1. Scaling with Problem Size

The scalability of the implementation for two different problem sizes are discussed below. The number of processor cores used to solve the medium sized problem are 1, 2, 4, 8, 12, 16, 20 and 24. The number of processor cores used for solving the larger problem are 12, 24, 48, 72 and 96. Note that the system utilizes 24 processor cores per computational node. The wall time taken for key operations are tracked and plotted to evaluate the parallel performance and identify potential bottlenecks.

HIGH LEVEL OPERATIONS: The parallel scaling of high level operations involved in the solution process with the number of processors is shown in Figure 14. The expected ideal scaling is overlaid as a black line. The forward and adjoint solutions show similar trends. The good scaling achieved for low number of processors deteriorates somewhat for larger number of processors. For the medium problem on the left, a slight performance degradation from 8 to 12 processors can be seen for both the forward and adjoint solution time. This is a result of the extra cost of the factorization over this range of processors, as shown in Figure 15.

SOLUTION OPERATIONS: Figure 15 shows the parallel scalability in terms of the total time taken for operations involved in the solution process. The key operations considered are:

1. *Factorization* of the Jacobian matrix of the linearized nonlinear system at each iteration of the nonlinear solver
2. *Assembly operations* for setting up the matrices and residual of the governing equations for a given set of state and design variables
3. *Applying the factorization* to solve for Newton updates

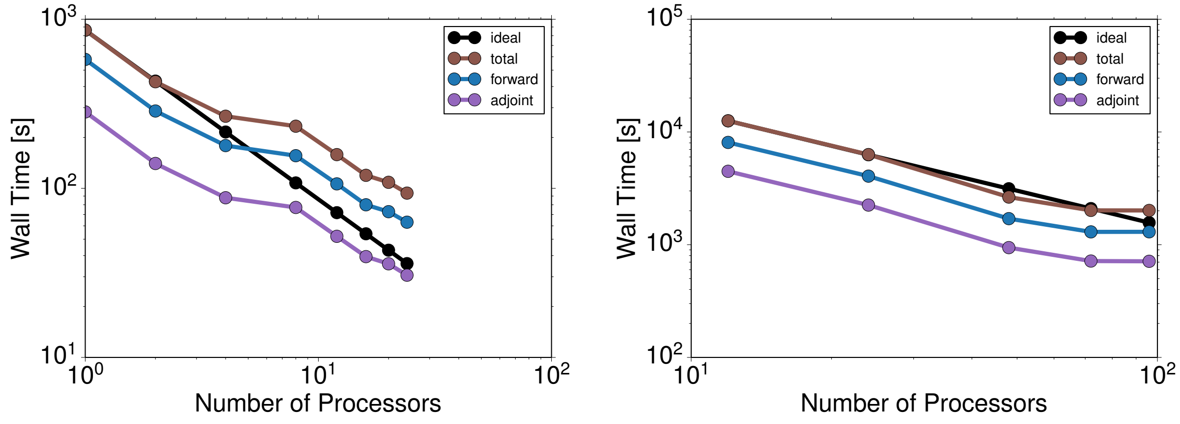
The matrix and residual assembly operations scale ideally. The matrix factorization and the application of matrix factorization scale well initially, but the performance suffers for larger number of processors.

ADJOINT OPERATIONS: The parallel scalability of the operations involved in the adjoint solution process is shown in Figure 16. The key operations considered are

1. *Factorization* of the transposed Jacobian matrices in the adjoint linear system of equations
2. *Assembly operations* for assembling the transposed matrices and contributions to the right hand side of the adjoint system of equations
3. *Applying the factorization* to solve for the adjoint variables for each functional of interest
4. *Matrix-vector products* involved in the equation for the evaluation of total derivative

Trends similar to the forward solution are also apparent in the adjoint solution for both problem sizes. The Jacobian-vector products scale ideally. The percentage of time taken for aforementioned operations are plotted in Figures 17 and 18. The matrix factorization takes the majority of computational time and emerge as the single most expensive operation. The implication is that the scaling of the matrix factorizations are crucial to the overall parallel performance of the framework. Another observation is that, the forward mode takes more time than the reverse mode under scenarios considered here. In general it follows that,

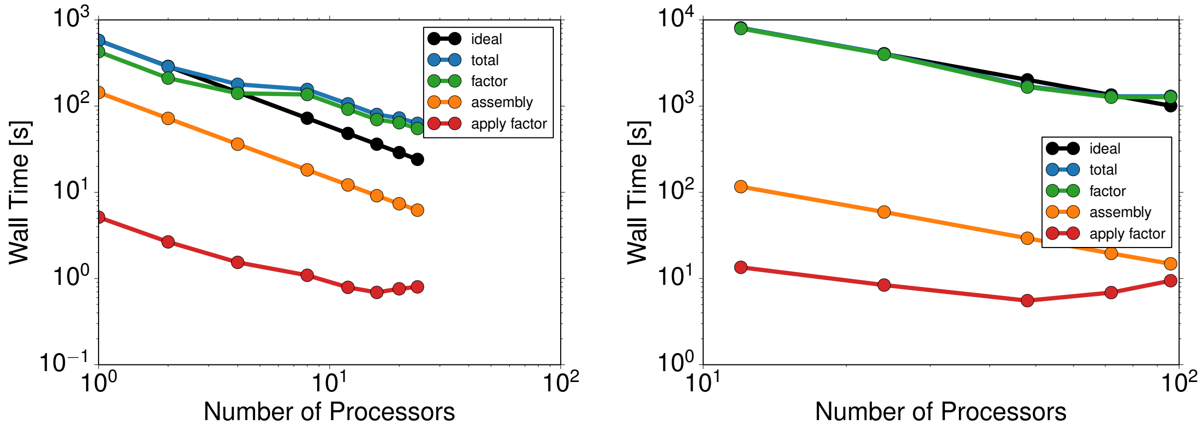
$$\text{time}_{\text{forward}} \approx \frac{\text{number of nonlinear iterations}}{\text{number of functionals of interest}} \times \text{time}_{\text{reverse}}. \tag{53}$$



(a) 192,000 degrees of freedom

(b) 2 millions degrees of freedom

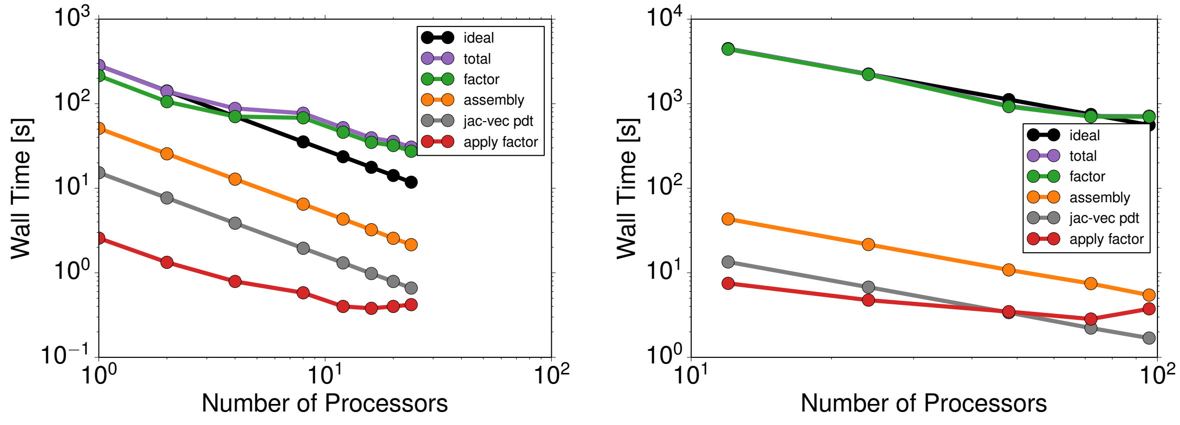
Figure 14: Parallel scalability assessments for high level operations.



(a) 192,000 degrees of freedom

(b) 2 million degrees of freedom

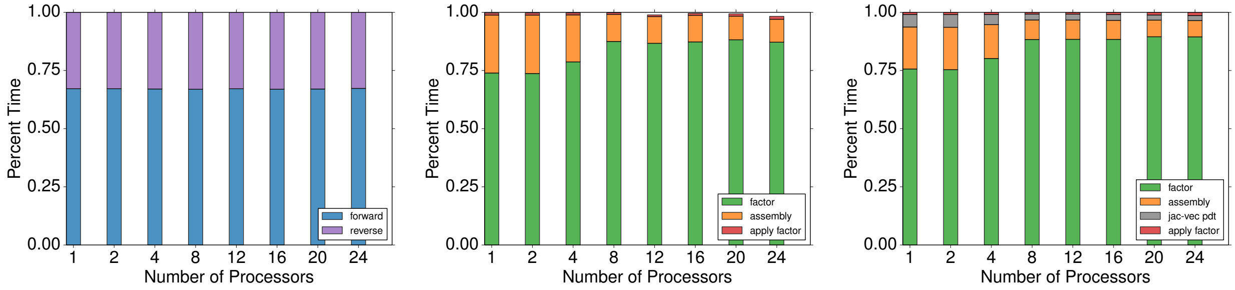
Figure 15: Parallel scalability assessments for forward mode operations.



(a) 192,000 degrees of freedom

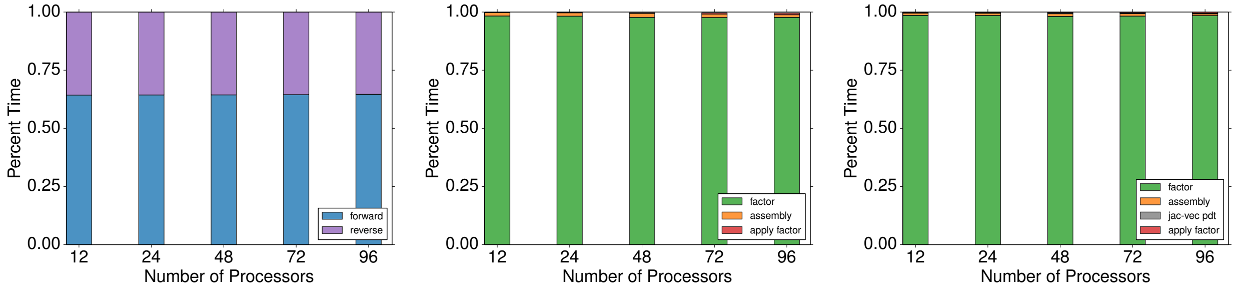
(b) 2 millions degrees of freedom

Figure 16: Parallel scalability assessments for the adjoint operations.



(a) Overall performance (b) Solution operations (c) Adjoint operations

Figure 17: Percentage of time taken for different operations for the model with 192, 000 degrees of freedom.

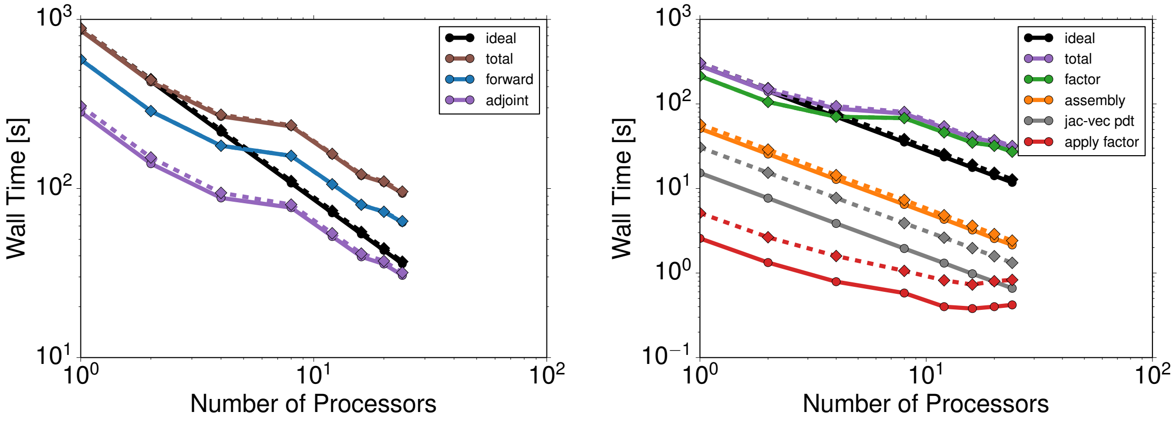


(a) Overall performance (b) Solution operations (c) Adjoint operations

Figure 18: Percentage of time taken for different operations for the model with 2 million degrees of freedom.

2. *Scaling with Number of Functionals*

The effect of increasing the number of functionals on the computational cost of the adjoint operations for the model with 192, 000 degrees of freedom is plotted in Figure 19. The total adjoint solution time increases by less than 5%, for an increase in the number of functionals. All the adjoint operations except the matrix factorization see an increase in time due to additional computations. This is because the same matrix factorization is applied to different right hand sides. Overall, the lines are shifted upwards without any visible changes to the trends shown before in Figure 16.



(a) Overall performance (b) Adjoint operations

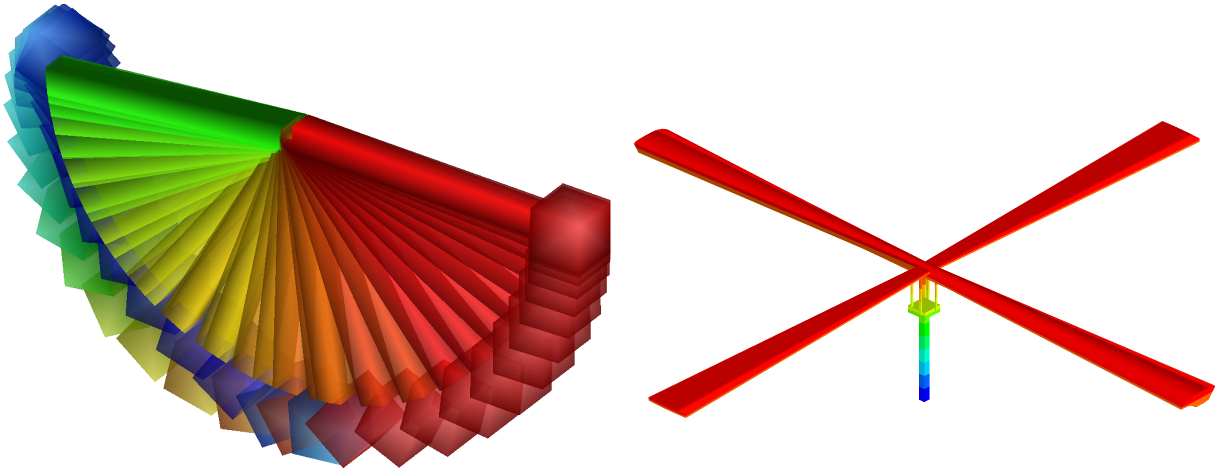
Figure 19: Parallel scalability assessments for increasing number of functionals of interest.

C. Flexible Multibody Dynamics

The illustration of flexible multibody dynamics capabilities of the framework with the simulation of rigid as well as flexible models follow.

1. Rigid Models

The multibody dynamics of pendulum and rotor models constructed using rigid bodies connected through spherical and revolute kinematic constraints are shown in Figure 20. Figure 21 shows the use of rigid-link between two cube-shaped rigid bodies. For this simulation, the angular velocity components are applied along all three coordinate axes. The contours represent the rotations in each body and it can be seen that the rotations are transmitted between the bodies with the use of rigid-link.



(a) double pendulum

(b) rotor

Figure 20: Time lapse plots of the configuration of rigid bodies connected through spherical and revolute constraints.

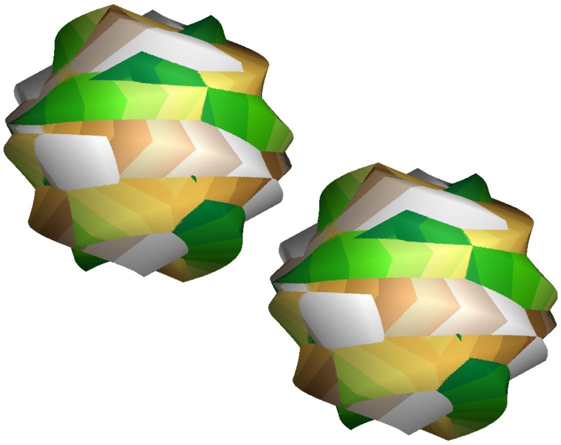
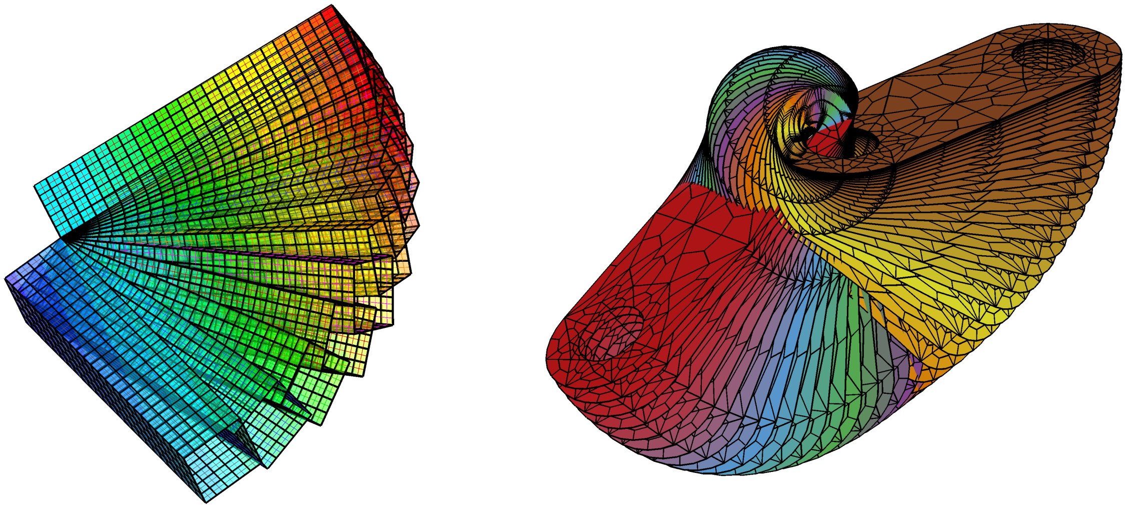


Figure 21: Contours of the rotations of two rigid bodies connected through a rigid-link.

2. Flexible Models

The flexible simulation models considered for study are (a) shell box, (b) punched plate and (c) rotor blade models. The unstructured finite-element meshes of the flexible models are generated through the open-source program *gmsh* [20]. Figure 22 shows the dynamics of a shell box and punched plate flexible models. Here, the translational degrees of freedom of one of the finite-element nodes are constrained, and the models are allowed to oscillate under the influence of gravity. The clamped rotor blade model shown in Figure 23 has $225,000$ degrees of freedom and is applied an initial angular velocity resulting in the shown configuration over time.



(a) shell box

(b) punched plate

Figure 22: Time lapse plots showing the configuration of flexible bodies.

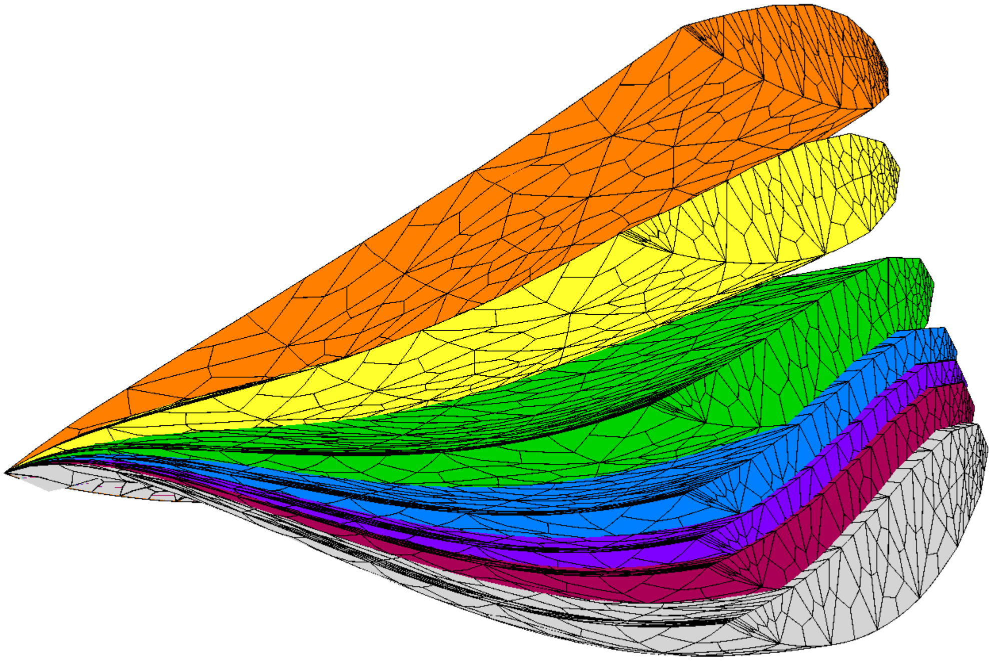


Figure 23: Time lapse plot showing the configuration of a clamped rotor blade model.

3. Heterogeneous Models

A schematic representation of connecting rigid and flexible bodies is shown in Figure 24. Each node, identified with a unique index, i, has a physical location, $\mathbf{x}_i$, associated with it. The constraint and link elements are associated with dummy nodes (2, 3, 4 and 5) that serve to preserve the generality of implementation aspects, rather than bearing a geometrical significance. The same treatment can be extended for linking multiple rigid and flexible bodies. Figure 25 shows the chaotic motion of a flexible plate rigidly linked with a rigid body, that is connected to a fixed-point in space through spherical constraint. The rigid body and the link are suppressed for visualization purposes.

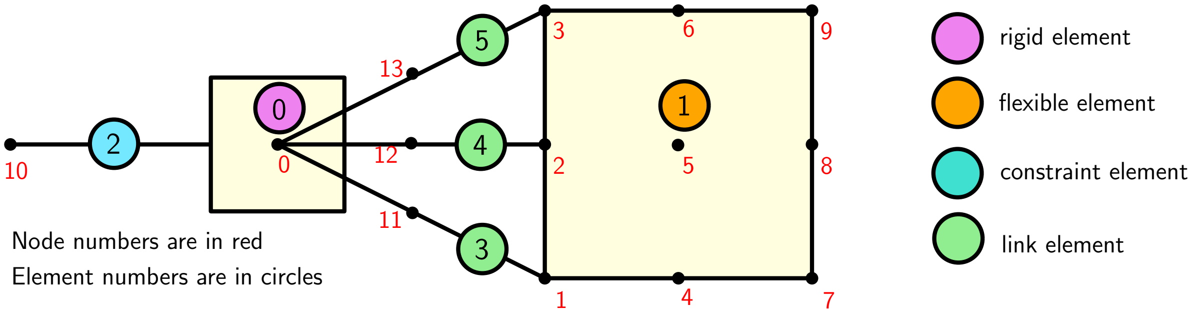


Figure 24: A schematic representation of connecting rigid and flexible elements using rigid-link.

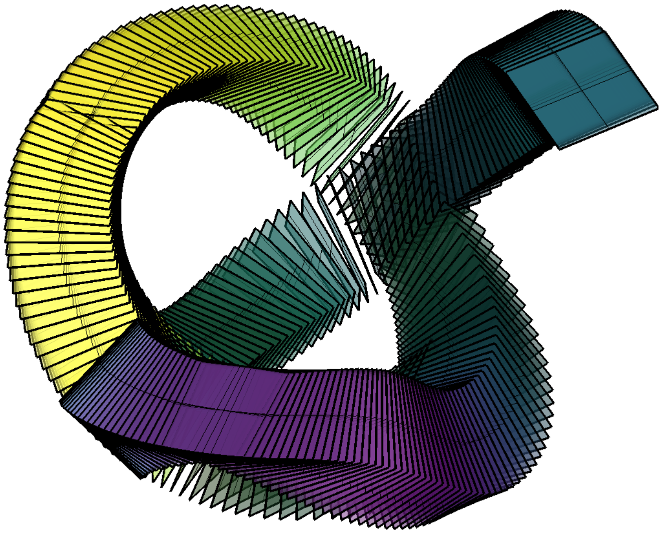


Figure 25: The configuration of a flexible plate connected to a rigid body (not shown) through a rigid-link.

IV. Conclusion

In this paper, we presented a framework for analysis and adjoint-based derivative evaluation for second-order systems, such as flexible multibody dynamics. The framework will provide the ground work required for multidisciplinary optimization of complex flexible multibody systems like rotorcraft. The availability of analysis and adjoint schemes for various time-marching methods facilitates the selection of solution methods tailored to the problem under investigation. The mathematical adjoint formulations are numerically verified with complex-step method and are shown to be accurate. The parallel scalability of the implementation will be useful for high-fidelity models with large number of degrees of freedom. The components central to the construction of full-fledged rotor models are demonstrated with simple simulations.

References

[1] H. M. Adelman and R. T. Haftka. Sensitivity analysis of discrete structural systems. *AIAA Journal*, 24(5):823–832, 1986. doi:10.2514/3.48671.

[2] M. A. Akgun, R. T. Haftka, K. C. Wu, J. L. Walsh, and J. H. Garcelon. Efficient structural optimization for multiple load cases using adjoint sensitivities. *AIAA Journal*, 39(3):511–516, 2001. doi:10.2514/2.1336.

[3] R. Alexander. Diagonally implicit Runge–Kutta methods for stiff O.D.E.'s. *SIAM J. Numer. Anal.*, 14(6):1006–1021, 1977. doi:10.1137/0714068.

[4] W. Anderson and V. Venkatakrishnan. Aerodynamic design optimization on unstructured grids with a continuous adjoint formulation. *Computers & Fluids*, 28(45):443 – 480, 1999. ISSN 0045-7930. doi:http://dx.doi.org/10.1016/S0045-7930(98)00041-3.

[5] F. Bashforth and J. C. Adams. An attempt to test the theories of capillary action by comparing the theoretical and measured forms of drops of fluid, with an explanation of the method of integration employed in constructing the tables which give the theoretical forms of such drops. Cambridge University Press, 1883.

[6] O. A. Bauchau. *Flexible Multibody Dynamics*. Springer Netherlands, 2011. ISBN 978-94-007-0334-6. doi:10.1007/978-94-007-0335-3.

[7] A. D. Belegundu and J. S. Arora. A sensitivity interpretation of adjoint variables in optimal design. *Computer Methods in Applied Mechanics and Engineering*, 48(1):81–89, 1985. ISSN 00457825. doi:10.1016/0045-7825(85)90068-4.

[8] K. Brenan, S. Campbell, and L. Petzold. *Numerical Solution of Initial-Value Problems in Differential-Algebraic Equations*. Society for Industrial and Applied Mathematics, 1995. doi:10.1137/1.9781611971224.

[9] G. W. Burgreen and O. Baysal. Three-Dimensional Aerodynamic Shape Optimization Using Discrete Sensitivity Analysis. *AIAA Journal*, 34, No. 9:1761–1770, 1996. doi:10.2514/3.13305.

[10] J. C. Butcher. Implicit Runge–Kutta processes. *Mathematics of Computation*, 18(85):50–64, 1964. ISSN 00255718, 10886842. doi:10.1090/S0025-5718-1964-0159424-9.

[11] G. D. Byrne and A. C. Hindmarsh. A polyalgorithm for the numerical solution of ordinary differential equations. *ACM Transactions on Mathematical Software (TOMS)*, 1(1):71–96, 1975. ISSN 0098-3500. doi:10.1145/355626.355636.

[12] Y. Cao, S. Li, L. Petzold, and R. Serban. Adjoint sensitivity analysis for differential-algebraic equations: The adjoint dae system and its numerical solution. *SIAM Journal on Scientific Computing*, 24(3):1076–1089, 2003. doi:10.1137/S1064827501380630.

[13] J. Cash. Diagonally implicit Runge–Kutta formulae for the numerical integration of nonlinear two-point boundary value problems. *Computers & Mathematics with Applications*, 10(2):123 – 137, 1984. ISSN 0898-1221. doi:10.1016/0898-1221(84)90043-9.

[14] C. F. Curtiss and J. O. Hirschfelder. Integration of Stiff Equations. *Proceedings of the National Academy of Sciences of the United States of America*, 38(3):235–243, 1952.

[15] J.-Y. Ding, Z.-K. Pan, and L.-Q. Chen. Second order adjoint sensitivity analysis of multibody systems described by differential–algebraic equations. *Multibody System Dynamics*, 18(4):599–617, 2007. ISSN 1573-272X. doi:10.1007/s11044-007-9080-4.

[16] D. Dopico, Y. Zhu, A. Sandu, and C. Sandu. Direct and adjoint sensitivity analysis of ordinary differential equation multibody formulations. *Journal of Computational and Nonlinear Dynamics*, 10(1):1–7, 2014. doi:10.1115/1.4026492.

[17] L. Fox and E. T. Goodwin. Some new methods for the numerical integration of ordinary differential equations. *Mathematical Proceedings of the Cambridge Philosophical Society*, 45(3):373–388, 007 1949. doi:10.1017/S0305004100025007.

[18] C. W. Gear. The automatic integration of ordinary differential equations. *Commun. ACM*, 14(3):176–179, Mar. 1971. ISSN 0001-0782. doi:10.1145/362566.362571.

[19] C. W. Gear. Simultaneous numerical solution of differential-algebraic equations. *IEEE Transactions on Circuit Theory*, 18 (1):89–95, Jan 1971. ISSN 0018-9324. doi:10.1109/TCT.1971.1083221.

[20] C. Geuzaine and J. F. Remacle. Gmsh: A 3-D finite element mesh generator with built-in pre- and post-processing facilities. *International Journal for Numerical Methods in Engineering*, 79(11):1309–1331, 2009. ISSN 00295981. doi:10.1002/nme.2579.

[21] B. Grossman, Z. Gürdal, R. T. Haftka, G. J. Strauch, and W. M. Eppard. Integrated aerodynamic/structural design of a sailplane wing. *Journal of Aircraft*, 25(9):855–860, 2013/09/28 1988. doi:10.2514/3.45980.

[22] R. Haftka and H. Adelman. Recent developments in structural sensitivity analysis. *Structural optimization*, 1(3):137–151, 1989. ISSN 0934-4373. doi:10.1007/BF01637334.

[23] P. Henrici. *Discrete variable methods in ordinary differential equations*. Wiley, New York, 1962.

[24] A. C. Hindmarsh. {ODEPACK}, A Systematized Collection of {ODE} Solvers. *IMACS Transactions on Scientific Computation*, 1:55–64, 1983.

[25] A. Jameson. Aerodynamic design via control theory. *Journal of Scientific Computing*, 3(3):233–260, 1988. ISSN 08857474. doi:10.1007/BF01061285.

[26] G. J. Kennedy and J. E. Hicken. Improved constraint-aggregation methods. *Computer Methods in Applied Mechanics and Engineering*, 289:332 – 354, 2015. ISSN 0045-7825. doi:10.1016/j.cma.2015.02.017.

[27] G. J. Kennedy and J. R. R. A. Martins. A parallel finite-element framework for large-scale gradient-based design optimization of high-performance structures. *Finite Elements in Analysis and Design*, 87(0):56 – 73, 2014. ISSN 0168-874X. doi:10.1016/j.finel.2014.04.011.

[28] G. Kreisselmeier and R. Steinhauser. Systematic control design by optimizing a vector performance index. In *International Federation of Active Controls Symposium on Computer-Aided Design of Control Systems*, Zurich, Switzerland, 1979.

[29] J. R. R. A. Martins, P. Sturdza, and J. J. Alonso. The complex-step derivative approximation. *ACM Transactions on Mathematical Software*, 29(3):245–262, Sept. 2003. doi:10.1145/838250.838251.

[30] J. R. R. A. Martins, J. J. Alonso, and J. J. Reuther. A coupled–adjoint sensitivity analysis method for high–fidelity aero–structural design. *Optimization and Engineering*, 6:33–62, 2005. doi:10.1023/B:OPTE.0000048536.47956.62.

[31] L. Meirovitch. *Principles and techniques of vibrations*. Prentice Hall, Upper Saddle River, NJ, 1st edition, 1997. ISBN 0023801417.

[32] F. R. Moulton. *New Methods in Exterior Balistics*. University of Chicago Press, 1926.

[33] K. Nachbagauer, S. Oberpeilsteiner, K. Sherif, and W. Steiner. The use of the adjoint method for solving typical optimization problems in multibody dynamics. *Journal of Computational and Nonlinear Dynamics*, 10(6):1–10, 2015. ISSN 1555-1415. doi:10.1115/1.4028417.

[34] N. M. Newmark. A method of computation for structural dynamics. *Journal of the Eng. Mech. Div.*, 85(3), 1959.

[35] P. Eberhard. Adjoint variable method for sensitivity analysis of multibody systems interpreted as a continuous, hybrid form of automatic differentiation. Proceedings of the Second International Workshop of Computational Differentiation, Santa Fe, NM, 1996.

[36] L. L. Sherman, A. C. Taylor III, L. L. Green, and P. A. Newman. First- and second-order aerodynamic sensitivity derivatives via automatic differentiation with incremental iterative methods. *Journal of Computational Physics*, 129:307 – 331, 1996. doi:10.2514/6.1994-4262.

[37] W. Squire and G. Trapp. Using complex variables to estimate derivatives of real functions. *SIAM Review*, 40(1):110–112, 1998. doi:10.1137/S003614459631241X.