A Nonlinear CFD/Multibody Incremental-Dynamic Model for A Constrained Mechanism

Seyed Mohammadali Rahmati¹ and Alireza Karimi²,*

¹ School of Biological Sciences, Georgia Institute of Technology, Atlanta, GA 30332, USA; smarahmati@gmail.com
² Department of Ophthalmology and Visual Sciences, University of Alabama at Birmingham, Birmingham, AL 35294, USA
* Correspondence: akarimi@uabmc.edu

Abstract: Numerical analysis of a multibody mechanism moving in the air is a complicated problem in computational fluid dynamics (CFD). Analyzing the motion of a multibody mechanism in a commercial CFD software, i.e., ANSYS Fluent®, is a challenging issue. This is because the components of a mechanism have to be constrained next to each other during the movement in the air to have a reliable numerical aerodynamics simulation. However, such constraints cannot be numerically modeled in a commercial CFD software, and needs to be separately incorporated into models through the programming environment, such as user-defined functions (UDF). This study proposes a nonlinear-incremental dynamic CFD/multibody method to simulate constrained multibody mechanisms in the air using UDF of ANSYS Fluent®. To testify the accuracy of the proposed method, Newton–Euler dynamic equations for a two-link mechanism are solved using Matlab® ordinary differential equations (ODEs), and the numerical results for the constrained mechanisms are compared. The UDF results of ANSYS Fluent® shows good agreement with Matlab®, and can be applied to constrained multibody mechanisms moving in the air. The proposed UDF of ANSYS Fluent® calculates the aerodynamic forces of a flying multibody mechanism in the air for a low simulation cost than the constraint force equation (CFE) method. The results could have implications in designing and analyzing flying robots to help human rescue teams, and nonlinear dynamic analyses of the aerodynamic forces applying on a moving object in the air, such as airplanes, birds, flies, etc.

Keywords: CFD/multibody mechanisms; nonlinear dynamics; user-defined functions; ANSYS Fluent®; constrained dynamic problems; Matlab®

1. Introduction

Analyzing the equations of motion for a multibody mechanism is an intricate issue in the field of dynamic mechanics, since several dynamic equations with high-order nonlinearities are essentially required to be solved [1,2]. If a multibody mechanism enlarges with some active and passive joints moving in the air, the difficulties of analysis of the system will be amplified due to the initial and boundary conditions of the model [3]. Such mechanisms can drift away from their joint connections during their movement in the air owing to the set of complicated aerodynamic forces, such as drag, shear, and normal [4].

Constraint force equation (CFE) has been widely used for dynamic analysis of multibody mechanisms [5–11]. In CFE, the constraint forces and momentums are calculated using multibody dynamic equations, and then incorporated into the model as a separate-external force on each part/component of a multibody mechanism. Although CFE is in wide using for various types of multibody mechanisms under different initial and boundary conditions, the main disadvantage of that is the external forces applying on a mechanism should be determined during the procedure as a known function and cannot be dependent on the spatial location of each part/component of a multibody mechanism. The constraint forces in the CFE method are computed based on the external forces, so the
values of external forces should be known independent of the spatial positions, which is in contrast with the usual performance of the ANSYS Fluent® software, where the external forces at each moment are estimated by the spatial position of the mechanism and complex numerical analysis corresponding to finite element method (FEM) and computational fluid dynamics (CFD) [12–14]. Therefore, using CFE seems costly as it needs to be performed in two different software, one for dynamic coupling, e.g., Matlab® (MathWorks, Natick, MA, USA) and one for solving aerodynamic equations and simulating the multibody system, e.g., ANSYS Fluent® (ANSYS Fluent® Software: CFD Simulation, Canonsburg, PA, USA). This is why the application of user-defined functions (UDF) is significant, as it is faster and more accurate than those approaches using so far, especially CFE. UDF is flexible and powerful, and allow to modify the behavior of ANSYS Fluent® to satisfy the particular modeling needs, such as modifications in the initial and boundary conditions, material models, and various physical transformations [15]. However, it should be noted that due to the complexity of the UDF programming, its application in analyzing the dynamics of a multibody mechanism has not been extended.

A problem corresponding to analyzing the dynamic separation of multiple bodies in the air is mostly related to store separation from aircraft [16,17], for example, the separation of the X-15 research vehicle from the B-52 carrier aircraft [18,19]. In such cases, the store and the X-15 vehicle are much smaller than that of the parent vehicle. However, there would be some other class of stage separation problem, which involves the separation of two vehicles of comparable sizes. In such cases, these would be defined as multi-stage reusable launch vehicles where the integrity of each stage is important after separation [20,21]. NASA conducted extensive research on stage separation of multi-stage reusable launch vehicles since the 1960s. These studies mostly focused on the problem of separation of generic two-stage reusable launch vehicles. However, NASA is always seeking to develop a better and more comprehensive stage separation tool that included wind tunnel testing, as well as the development and validation of CFD and engineering level simulation tools [22,23].

With these considerations in mind, a stage separation analysis and simulation tool was developed called Conceptual Separation, which is already implemented into a commercial multibody dynamic analysis software ADAMS®. However, a disadvantage of Conceptual Separation would be that it cannot be simply integrated into standard trajectory simulation software for performing efficient and seamless end-to-end simulations of launch vehicle trajectories because it is tied to ADAMS [4,18]. Therefore, the development of a new methodology is required to be implemented into a two-link mechanism.

This study was aimed to propose a nonlinear CFD/multibody incremental-dynamic method to simulate a constrained two-link mechanism moving in the air. The simulations were conducted on the basis of Newton–Euler equations and the extracted equations were then input to ANSYS Fluent® software as the UDF. To check the accuracy of the UDF, the coupled-CFD equations were solved in Matlab® software using fourth order Runge–Kutta method [24] for an arbitrary external force field, and the outcomes were compared to the UDF results of ANSYS Fluent®, which were solved by forward finite-difference. Since the fourth order Runge–Kutta method produces results with an error of order 4 and forward finite-difference leads to the error order 1, the method applied in Matlab® is considered as a benchmark test for the UDF results of ANSYS Fluent®. An arbitrary external force field was selected because, unlike ANSYS Fluent®, Matlab® does not have a solver for aerodynamic equations, so it cannot compute aerodynamic forces. Then the verified method was applied for a real aerodynamic force field using UDF of ANSYS Fluent®. To the best of the authors’ knowledge, since ANSYS Fluent® is unable to analyze the dynamics of a CFD/multibody mechanism under the constraint forces and momentums, the proposed method in this study can well incorporate into the software to accurately analyze the nonlinear dynamics of any complex multibody mechanisms, i.e., flying robots, airplanes, birds, flies, under various initial and boundary conditions. Eventually, the dynamic and aerodynamic equations were coupled in UDF, as the proposed method simulates the whole components of a multibody system straightforward and does not need to apart the components for simulation.
2. Materials and Methods

A new form of governing equations is required to apply the algorithm of the problem-solving process. To obtain this new form, three stages need to be done, including the derivation of the governing equations based on the Newton–Euler dynamics, the time-discretization of velocities and accelerations, and rewriting the governing equations based on the generalized coordinates at the new moment of \((t_{n+1})\). The rewritten equations are the function of generalized coordinates and external aerodynamic forces at the present moment of \((t_n)\). By using UDF, the problem is formulated such that the values of aerodynamic forces computed by ANSYS Fluent® at \(t_n\) are called in the UDF and replaced as the inputs of the rewritten equations. Therefore, the spatial position of the bodies at the new time-step of \((t_{n+1})\), which is introduced by the generalized coordinates, is calculated. Then, the bodies in ANSYS Fluent® environment are transmitted to their new positions by UDF coding, and this process continues until the last time-step.

2.1. General Problem-Solving Algorithm

The dynamic equations of motion of the bodies were obtained based on the inertial, centrifugal/Coriolis, gravitational, and aerodynamic forces acting on the center of gravity (COG) of the bodies. Then, the equations of motion are discretized, as displayed in Figure 1. The method of multibody system dynamics was used to analyze the dynamics of constrained bodies, and ANSYS Fluent® software employed for the aerodynamic analysis.

![Figure 1. The general algorithm of problem-solving.](image)

Once the dynamic equations are derived, they can be introduced in the general form presented by Equation (1) as follows:

\[
M(y)\dot{y} + C(y, \dot{y}) + G(y) = R
\]  

(1)

where \(y\) is the vector of generalized coordinates corresponding to the number of system degrees of freedom; \(M(y)\) is the symmetric, positive, and definite mass matrix; \(C(y, \dot{y})\) is the vector of generalized centrifugal and Coriolis forces; \(G(y)\) is the vector of generalized gravitational forces; and \(R\) is the vector of generalized aerodynamic forces. Therefore, the system dynamics can be represented by the generalized coordinates \((y)\), their first and second derivatives \((\dot{y}, \ddot{y})\), and the external forces \((R)\), which are equal to the aerodynamic forces.
forces. The equations were discretized by the Euler method and represented by the
generalized coordinates at times of \( t_n \) and \( t_{n+1} \), and the aerodynamic forces at time of \( t_n \).
Then, the new system of equations was solved for the generalized coordinates at the new
time step (\( y_{n+1} \)) and represented as a function of generalized coordinates and forces at the
previous time step (\( y_n, R_n \)). At the initial moment (\( n = 0 \)), the spatial position, velocity
of the system, and aerodynamic forces are known, i.e., the initial condition is determined.
Therefore, by replacing the initial condition in the new system of equations, it is possible
to compute the generalized coordinates at the new time-step incrementally. The process
continued until the desired time level was achieved. The coding of this algorithm was
written in the UDF of ANSYS Fluent®.

To represent Equation (1) at the time-step of \( n + 1 \) as a function of previous time-step
(\( n \)), the forward finite-difference has been used, as follows:

\[
y_n = \frac{y_{n+1} - y_n}{\Delta t} + O(\Delta t) \quad (2)
\]

\[
y_n = \frac{y_{n+2} - 2y_{n+1} + y_n}{\Delta t^2} + O(\Delta t) \quad (3)
\]

where \( \Delta t \) is the time interval. A more precise numerical solution requires smaller time
intervals. Since the initial condition includes the initial position and velocities, at the time
of \( t = 0 \), i.e., \( n = 0 \) the values of \( y_0 \) and \( \dot{y}_0 \) are known. Therefore, the value of \( y_1 \) can be
computed by Equation (2). Finally, the values of generalized coordinates in the new time
step (\( y_2 \)) can be computed by replacing Equations (2) and (3) into Equation (1) and solving
Equation (1) for \( y_2 \). Similarly, the entire trajectory of the motion can be calculated.

Since the discretization was implemented for the coding process in ANSYS Fluent®
software, and the outputs of this software in the UDF environment are velocities and not
positions, the discretization was carried out for the velocities of generalized coordinates,
as follows:

\[
y_n = \frac{\dot{y}_{n+1} - \dot{y}_n}{\Delta t} + O(\Delta t) \quad (4)
\]

Therefore, Equation (1) should be solved for \( \dot{y}_{n+1} \) as a function of \( y_n, \dot{y}_n \), and \( R_n \); and
the same incremental procedure was applied.

2.2. Algorithm and Problem-Solving for a Four-Degree of Freedom System

A practical example of a planner system with four-degree of freedom based on the
algorithm is shown in Figure 1. All the steps of dynamic and aerodynamic analyses and
coding in the UDF environment were investigated. This example approves the credibility
of the algorithm.

To emphasize the proposed algorithm, a simple model was chosen. This model
includes two bodies with the masses of \( m_1 = 2 \) kg and \( m_2 = 2 \) kg, the lengths of \( l_1 = 0.5 \) m
and \( l_2 = 0.5 \) m, the center of gravity (COG) lengths of \( l_{o1} = 0.25 \) m and \( l_{o2} = 0.25 \) m, and
the principal inertias of \( I_1 = 1.4 \) kg\cdot m² and \( I_2 = 1.4 \) kg\cdot m². For the CFD solution, the
bodies were imposed to the air flow with velocity of \( \dot{v} = -3 \) m/s and \( \ddot{v} = -5 \) m/s for
constrained and unconstrained systems, respectively. These bodies were connected to each
other by a revolute joint. Due to the flow of fluid, aerodynamic forces and torques were
applied to the COGs as presented in Figure 2a. In this figure, the vectors of \( [X_{10}/20, Y_{10}/20] \),
\( [X_{11}/21, Y_{11}/21] \), and \( [X_{12}/21, Y_{12}/21] \) show the origin, insertion, and COG of the links 1 and 2,
respectively. The vectors of \( R_1 \) and \( R_2 \) demonstrate the external aerodynamic forces acting
on the links of 1 and 2, respectively. The values of \( d_1 \) and \( d_2 \) are the distances between
the applied external forces and the COGs of links 1 and 2. It is desired to compute the
planer configurations of the two bodies, in which the following equality is satisfied during
the simulation:

\[
\begin{bmatrix}
X_{11} \\
Y_{11}
\end{bmatrix} = \begin{bmatrix}
X_{20} \\
Y_{20}
\end{bmatrix}
\] (5)
The satisfaction of Equation (5) would guarantee the connection of two bodies as a revolute joint.

The dynamic equations of the motion can be derived using the Newton–Euler method for the free body diagram of the system as indicated in Figure 2b. In this figure, the external forces of $R_1$ and $R_2$ are transmitted to the COGs for bodies 1 and 2, respectively. This creates the torques of $\tau_1$ and $\tau_2$. The external forces were rewritten based on their components $(R_{1x}, R_{1y}, R_{2x}$ and $R_{2y})$. The values of $F_{ex}$ and $F_{ey}$ are the internal forces in the directions of $x$ and $y$, and the variables of $\theta_1$ and $\theta_2$ are the angles of bodies 1 and 2 with respect to the horizon, respectively. The system of equations of motion for Figure 2 is presented as follows:

\[
R_{1x} + R_{2x} - \left(\tilde{X}_1 + l_1 f_1 - l_2 f_2\right) = m_1 \ddot{X}_1
\]
\[
R_{1y} + R_{2y} - m_2 \left(\tilde{Y}_1 - l_1 h_1 + l_2 h_2 + g\right) - m_1 g = m_1 \ddot{Y}_1
\]
\[
(q_1 \sin(\theta_1) - q_2 \cos(\theta_1)) \dot{l}_1 + \tau_1 = l_1 \ddot{\theta}_1
\]
\[
(-q_1 \sin(\theta_2) + q_2 \cos(\theta_2)) \dot{l}_2 + \tau_2 = l_2 \ddot{\theta}_2
\]

where:

\[
f_1 = \dot{\theta}_1 \sin(\theta_1) + \dot{\theta}_1^2 \cos(\theta_1)
\]
\[
f_2 = \dot{\theta}_2 \sin(\theta_2) + \dot{\theta}_2^2 \cos(\theta_2)
\]
\[
h_1 = \dot{\theta}_1 \cos(\theta_1) - \dot{\theta}_1^2 \sin(\theta_1)
\]
\[
h_2 = \dot{\theta}_2 \cos(\theta_2) - \dot{\theta}_2^2 \sin(\theta_2)
\]
\[
q_1 = R_{2x} - \tilde{X}_1 - l_1 f_1 + l_2 f_2
\]
\[
q_2 = R_{2y} - m_2 \left(\tilde{Y}_1 - l_1 h_1 + l_2 h_2 + g\right)
\]

If the external forces and torques were determined, there are four unknown variables (generalized coordinates), including $X_1$, $Y_1$, $\theta_1$ and $\theta_2$, which can be computed by solving the system of motion Equations (6)–(9).

The dynamics governing Equations (6)–(9) can be generally solved by two different numerical methods. One is to use Runge–Kutta fourth order to solve the system of ordinary differential equations (ODEs). To use this method, the external forces as known functions of time and generalized coordinates are required. Since in this study, an incremental solution has been adopted due to the dependency of spatial configurations on the aerodynamic forces and the nature of the numerical solution of CFD, another method is proposed. The alternative method is based on the discretization of the system dynamics and reformulation of the equations of motion by the algorithm presented in Figure 1. The first derivative of generalized coordinates at $t_{n+1}$ can be calculated by using Equation (4) for the discretization of the second derivatives of generalized coordinates and solving Equations (6)–(9) for
the time-step of \( n + 1 \). The system of equations was solved by Wolfram Mathematica® software (Wolfram Research, Champaign, IL, USA), and the code in detail are described in Supplementary file A (Solving of the equations of the motions). The code itself is also attached to the manuscript for the readers’ information.

To compare the results of the proposed method, the problem was solved by both methods for an arbitrary external force field, and the results were compared. The Matlab® ODE was applied for the first method and the ANSYS Fluent® environment using UDF for the second one. The external force field in detail is presented in Supplementary file B (Arbitrary external force field).

3. Results

The streamlines of the multibody mechanisms in the constrained and unconstrained conditions at different simulation times are illustrated in Figure 3. In the unconstrained condition, there were no constrained forces acting on the mechanisms’ joint, and the system was moved only under the aerodynamic forces from the fluid, which applies to the mechanism from right to left. Both bodies 1 and 2 at the beginning of the simulation were placed at the same displacements and angles in respect to each other. The results revealed that the unconstrained mechanisms due to their body weight distributions, which act at their centers, tend to change their alignment into the vertical position. The spatial position of body 2 as the aerodynamic forces act from right to left also influences the position of body 1 in a way that at the simulation time of 0.48 s, body 1 rotates, and consequently brings about a considerable distance between two bodies of 1 and 2. However, when it comes to the constrained mechanisms, the most important point is that two bodies kept their inner distance until the end of the simulation time. Another point is that the role of the spatial location of body 2 here does not become dominant in changing the location of body 1. It implies that in a constrained system, the bodies show more realistic moves under the aerodynamic forces, which is preferable in CFD simulations.

The contours of the velocity of the multibody mechanisms in the constrained and unconstrained conditions at different simulation times are illustrated in Figure 4. In the unconstrained condition, since the static pressure is placed in front of body 2 due to its vertical location, its angle is increased by the passage of simulation time, and it led to a conversion of pressure in the front and back sides of the body. Regarding body 1, after changing its spatial condition to horizontal, the same pressure conversions in both the front and back sides of this body occurred. This issue is observable at the simulation times of 0.48 s and 1.14 s between the unconstrained and constrained bodies of 1 and 2. At the simulation time of 0.48 s, in the unconstrained condition, the highest velocities were at the end of bodies 1 and 2, while in the constrained condition, the highest velocities were at the joint of bodies 1 and 2. This implies that in unconstrained mechanisms, the distribution of velocities were dramatically affected by the aerodynamic forces acting on the system, whereas in a constrained system, the joint can well manage the aerodynamic forces following by a realistic drop mode analyses.

The aerodynamics forces and torques acted on the center of gravities of bodies 1 and 2 in the X and Y directions for both the unconstrained and constrained conditions are presented in Figure 5. Under the aerodynamic forces in the unconstrained condition, at the simulation times of 0.6 s for body 1 and 0.25 s for body 2, the motion of the mechanism was changed. In these simulation times, the bodies were at the minimum velocity values. By changing the directions of the force vectors, the bodies of 1 and 2 relatively deviate to left, which is related to the forces in Y direction. These forces are always positive with the maximum value of 20 N. Meanwhile, as the gravity force of the bodies is \(-20 \text{ N}\), the summation of these two forces led to a negative acceleration in Y direction for the unconstrained mechanism. In the constrained mechanism, in most of the simulation time the forces in X direction were negative, which is related to the direction of the aerodynamic forces acting on the mechanism. When these forces are being compared to those of unconstrained mechanism, their values are two times bigger, which related to the
velocity of the flow here (−5 m/s compared to −3 m/s for the unconstraint mechanism). However, the forces in the Y direction were not altered (~20 N). The momentums here with the passage of simulation time were altered from positive to negative values, which address the rotation of the bodies.

![Figure 3. Streamlines of the (a,c,e) unconstrained and (b,d,f) constrained mechanisms at different simulation times.](image)

The X and Y forces acting on the joint of bodies 1 and 2 of the constrained mechanism are displayed in Figure 6. These forces were calculated by derivation of the $V_x$ and $V_y$ (velocities) at the center of gravities of bodies 1 and 2 and eventually the $A_x$ and $A_y$ (accelerations). The positive value of $F_x$ implies that the force acts from left to right, while the negative value shows that the force acts from right to left at the joint. The results revealed that the motion of body 2 is dominant and body 1 just is following that. The positive value of $F_y$ at the center of gravity of body 1 implies that the force acts from the down to up. However, since $F_y$ is almost negative during the simulation time it means that
body 1 experiences a higher force from the upper side, which leads to a rotation in that and take it to an upper right position compared to body 2.

Figure 4. Contours of the velocity of the (a,c,e) unconstrained and (b,d,f) constrained mechanisms at different simulation times.

To verify the developed UDF code, an arbitrary flow moving from right to left (Supplementary file B: Arbitrary external force field) was simulated using Matlab® ODE, and the results were compared to that of the UDF of ANSYS Fluent®. Since in Matlab® ODE the problem was simulated on the basis of the fourth order Runge–Kutta, the resulted error was three orders smaller than that of the UDF. The axial and angular velocities for both bodies 1 and 2 were computed using UDF and Matlab® ODE, and the results and their deviations are plotted in Figure 7. The mean percentage errors between the UDF and Matlab® ODE for the $V_x$, $V_y$, and $\omega$ of body 1 were 1.46%, 0.84%, and 1.87%, respectively and for body 2 were 1.15%, 0.82%, and 0.43%, respectively. These trivial errors imply the appropriateness of UDF code in solving a complex multibody mechanism.
Since in Matlab®, the problem was simulated on the basis of the ANSYS Fluent® and Mathematica® which address the rotational and constrained mechanisms, bodies 1 and 2.

Figure 5. Aerodynamic (a–d) X forces, (e–h) Y forces, and (i–l) momentums of the unconstrained and constrained mechanisms, bodies 1 and 2.

Figure 6. The constrained joint forces in the (a) X and (b) Y directions.

Selection of the time step size is one of the most important factors in the numerical integration of differential equation systems. To determine the appropriate size of the time step, first, the problem was solved by Matlab® ODE for an arbitrary force field with high frequencies (Supplementary file B: Arbitrary external force field) compared to a real force field. The method of ODE45 was selected. All ODE solvers provided by Matlab® employ a variety of variable-step methods. The Matlab® ODE solvers apply these methods by taking a step, computing the error at this step, checking if the value is greater than or less than the tolerance, and changing the step size accordingly. Utilizing a variable step guarantees that a large step size is utilized for low frequencies and a small step size is utilized for high frequencies. The ODE solvers within MATLAB are improved for a variable step, run faster with a variable step size, and clearly, the outcomes are more accurate. After solving the problem using ODE45 the smallest time step size was selected, and the differential equations of motion Equations (6)–(9) were solved by forward finite-difference Equations (2) and (3) in UDF environment to be sure that the error is less than the tolerance. The problem was solved for both unrealistic and realistic force fields. The reason a variable time step size was not selected for UDF, was that the ANSYS Fluent® would not accept variable values for time intervals, but only fixed step size. The proposed methodology can be described in short as computing equations of motion in the presence of external and internal forces applying on a mechanism as aerodynamic and constraint forces, respectively. Then, discretizing and reformulating them based on the next time step. Therefore, the next positions can be estimated by the previous values of positions and forces. The difficult part is to compute the equations of motion. Discretizing and reformulating the equations can be simply done by the Matlab® and Mathematica®. Applying the new discretized and
reformulated equation in UDF can be a little difficult [25]. Even though UDF programming is difficult, the physical and mathematical equations used are still based on Newton’s law and can be discretized.

![Figure 7](image_url)  
**Figure 7.** The comparison of UDF and Matlab® ODE, including (a) velocity in X direction of body 1, (b) velocity in Y direction of body 1, (c) angular velocity of body 1, (d) velocity deviation between UDF and Matlab® ODE in X direction of body 1, (e) velocity deviation between UDF and Matlab® ODE in Y direction of body 1, (f) angular velocity deviation between UDF and Matlab® ODE of body 2, (g) velocity in X direction of body 2, (h) velocity in Y direction of body 2, (i) angular velocity of body 2, (j) velocity deviation between UDF and Matlab® ODE in X direction of body 2, (k) velocity deviation between UDF and Matlab® ODE in Y direction of body 2, and (l) angular velocity deviation between UDF and Matlab® ODE of body 2.

**Limitations**

The study is limited by the following considerations. First, a 2D model was used to validate the accuracy of the proposed algorithm, which might not be a real case of a flying object in the air. However, the proposed approach can simply be extended to a 3D model, which will be the subject of a future report. Second, the solution was fulfilled through a stepwise approach. In this approach, the forces that are being calculated at each step are being used to calculate the location of the links at one-step ahead of the time. However, this could impose a step error into the simulation results, having a very small Δt would minimize the error in the simulation results. The experimental validation of this work was also a challenge for us due to the lack of enough resources and instruments. This is the authors’ objective to extend the code for a 3D model and experimentally verify which all will be the subject of a future report. The UDF code of the proposed algorithm is freely available to use as supplementary material (Supplementary file C: UDF code) and can extend our understanding of the mechanisms of the flying objects in the air.

**4. Conclusions**

This study proposed a new UDF coding to analyze the dynamics of a multibody mechanism with different degree-of-freedom in ANSYS Fluent® software. We could
address the shortcomings of the proposed models so far, especially CFE, as it unable to model a multibody mechanism in the presence of various aerodynamics loadings related to the spatial configurations of the bodies as fulfilled in ANSYS Fluent® software. The mean percentage errors between the UDF and Matlab® ODE for the \(V_x\), \(V_y\), and \(\omega\) of body 1 were found to be 1.4%, 0.8%, and 1.8%, respectively, and for body 2 were 1.1%, 0.8%, and 0.4%, respectively. The negligible errors found here represent the powerfulness of the proposed UDF code in addressing the complex mechanical response of a multibody mechanism flying in the air. In addition, the proposed models so far have used coupling of ANSYS Fluent® and Matlab® to deal with the equations; however, the simulation time in this case dramatically increases. The proposed method can directly address this issue using a single programming code, namely, UDF of ANSYS Fluent®. In this method, the equations of motion were discretized on the basis of the velocities and accelerations of a multibody mechanism under the aerodynamic forces. The equations were solved incrementally according to the location of the rigid body and its aerodynamic forces acting on it. The equations were first solved in ANSYS Fluent® and then updated at each increment in the UDF code. The proposed method here has implications in analyses of flying robots, airplanes, birds, flies, etc., to provide a more accurate dynamic concept of their movements and induced forces during their flies.

Supplementary Materials: The following are available online at https://www.mdpi.com/2076-3417/11/3/1136/s1, Supplementary file A: Solving of the equations of the motions, Supplementary file B: Arbitrary external force field, Supplementary file C: UDF code.

Author Contributions: Conceptualization, S.M.R. and A.K.; methodology, S.M.R. and A.K.; software, S.M.R. and A.K.; validation, S.M.R. and A.K.; formal analysis, S.M.R. and A.K.; investigation, S.M.R. and A.K. Both authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References
1. Costeira, J.; Kanade, T. A multi-body factorization method for motion analysis. In Proceedings of the IEEE Fifth International Conference on Computer Vision, Washington, DC, USA, 20–23 June 1995.
2. De Jalon, J.G.; Bayo, E. Kinematic and Dynamic Simulation of Multibody Systems: The Real-Time Challenge; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2012.
3. O’Brien, J.F.; Wen, J.T. Redundant actuation for improving kinematic manipulability. In Proceedings of the IEEE International Conference on Robotics and Automation, Detroit, MI, USA, 10–15 May 1999.
4. Toniolo, M.; Tartabini, P.; Pamadi, B.; Hotchko, N. Constraint force equation methodology for modeling multi-body stage separation dynamics. In Proceedings of the 46th AIAA Aerospace Sciences Meeting and Exhibit, Reno, NV, USA, 7–10 January 2008; American Institute of Aeronautics and Astronautics: Reno, NV, USA, 2008.
5. Kamman, J.W.; Huston, R. Dynamics of constrained multibody systems. J. Appl. Mech. 1984, 51, 899–903. [CrossRef]
6. Shabana, A.A. Flexible multibody dynamics: Review of past and recent developments. Multibody Syst. Dyn. 1997, 1, 189–222. [CrossRef]
7. Marques, F.; Souto, A.P.; Flores, P. On the constraints violation in forward dynamics of multibody systems. Multibody Syst. Dyn. 2016, 39, 385–419. [CrossRef]
8. Liu, F.; Zhang, J.; Hu, Q. A modified constraint force algorithm for flexible multibody dynamics with loop constraints. Nonlinear Dyn. 2017, 90, 1885–1906. [CrossRef]
9. Pappalardo, C.M.; Guida, D. On the Lagrange multipliers of the intrinsic constraint equations of rigid multibody mechanical systems. Arch. Appl. Mech. 2017, 88, 419–451. [CrossRef]
10. Peterson, D.L.; Gede, G.; Hubbard, M. Symbolic linearization of equations of motion of constrained multibody systems. Multibody Syst. Dyn. 2014, 33, 143–161. [CrossRef]
11. Lee, H.-W.; Roh, M.-I. Review of the multibody dynamics in the applications of ships and offshore structures. Ocean Eng. 2018, 167, 65–76. [CrossRef]
12. Tartabini, P.V.; Roithmayr, C.M.; Toniolo, M.D.; Karlgaard, C.D.; Pamadi, B.N. Modeling Multibody Stage Separation Dynamics Using Constraint Force Equation Methodology. J. Spacecr. Rocket. 2011, 48, 573–583. [CrossRef]
13. Tartabini, P.; Roithmayr, C.; Karlgaard, C.; Toniolo, M.; Pamadi, B. Verification of a Constraint Force Equation Methodology for Modeling Multi-Body Stage Separation. In Proceedings of the AIAA Modeling and Simulation Technologies Conference and Exhibit, Guidance, Navigation, and Control and Co-located Conferences: American Institute of Aeronautics and Astronautics, Honolulu, HI, USA, 18–19 August 2008.

14. Pamadi, B.N.; Toniolo, M.D.; Tartabini, P.V.; Roithmayr, C.M.; Albertson, C.W.; Karlgaard, C.D. Development of Constraint Force Equation Methodology for Application to Multi-Body Dynamics Including Launch Vehicle Stage Separation; Langley Research Center: Hampton, VA, USA, 2016.

15. Fluent, A. Ansys Fluent UDF Manual; ANSYS Inc.: Canonsburg, PA, USA, 2015.

16. Xue, F.; Wang, Y.; Qin, H. Derivation and validation of wind tunnel free-flight similarity law for store separation from aircraft. Aerosp. Sci. Technol. 2020, 97, 105614. [CrossRef]

17. Xue, F.; Qin, H.; Wang, Y.; Jiang, Z.; Jin, X.; Bai, P. Free flight wind wind tunnel test similarity law derivation for light store separation from aircraft. Adv. Mech. Eng. 2019, 11. [CrossRef]

18. Albertson, C.; Tartabini, P.; Pamadi, B. End-to-end simulation of launch vehicle trajectories including stage separation dynamics. In Proceedings of the AIAA Atmospheric Flight Mechanics Conference, Toronto, ON, Canada, 13 August 2012; p. 4863.

19. Pamadi, B.N.; Neirynck, T.A.; Hotchk, N.J.; Scallion, W.I.; Murphy, K.J.; Covell, P.F.; Hotcko, N.J. Simulation and analyses of stage separation of two-stage reusable launch vehicles. J. Spacecr. Rocket. 2007, 44, 66–80. [CrossRef]

20. Rasky, D.; Pittman, R.B.; Newfield, M. The reusable launch vehicle challenge. Space 2006, 7208. [CrossRef]

21. Briese, L.E.; Schnepper, K.; Acquatlata, P. Advanced modeling and trajectory optimization framework for reusable launch vehicles. In Proceedings of the IEEE Aerospace Conference, Big Sky, MT, USA, 3 March 2018; pp. 1–18.

22. Pallegoix, J.F.; Barroca, B.; Brenner, P. Use of CFD for Fine Aerodynamic Characterisation of A Launcher. ESASP 2011, 692, 99.

23. Murphy, K.; Scallion, W. Experimental Stage Separation Tool Development in NASA Langley’s Aerothermodynamics Laboratory. In Proceedings of the AIAA Atmospheric Flight Mechanics Conference and Exhibit, San Francisco, CA, USA, 1 January 2005; p. 6127.

24. Dormand, J.R.; Prince, P.J. A family of embedded Runge-Kutta formulae. J. Comput. Appl. Math. 1980, 6, 19–26. [CrossRef]

25. Chen, Q.; Hsu, M.; Zhang, B.; Wu, R. Multi-Input, Multi-Output-Per-Input User-Defined-Function-Based Database Operations. United States Patent US 8,805,870; Hewlett Packard Development Co LP Assignee, Washington, DC, USA, 12 August 2014.