I R I F sa
A a E g g M S s s

Wenqin Zhou[1], David J. Je rey[1], Gregory J. Reid[1], Chad Schmitke[2], and John McPhee[2]

[1] n h , h nv n n , n n, n , n 5
[2] n n n n , nv ,
, n , n 2 3 1

A s c . h RifSimp n n - n u n u nv u v . h b h n RifSimp h n n -w b un n n - n n n n n . RifSimp w u n n n h n u u b . h u n w n b h Dynaflex, wh h n u h- n u b h n n n n u n w h b n n . n h n RifSimp u u h n b u - n n u un u n , n n f h u n n b . u h n h , u n n - u nv u v n v n u h b . h RifSimp b h b nv n . n n u h mu n n h u b -

For multibody systems, the general form of the dynamic equations is

$$M(\ ,\ ,\dot{\ })\ddot{\ } + \ {}^{T}_{q} \ = F(\ ,\ ,\dot{\ }) , \tag{1}$$

$$(\ ,\ ) = 0 . \tag{2}$$

Here, is a vector of generalized co-ordinates, $M(\ ,\ ,\dot{\ })$ is the mass matrix, is a vector of the constraint equations and is a vector of Lagrange multipliers [17, 18]:

$$= \left[\begin{array}{l} {}^{1} \\ \\ \end{array} \right.$$

11 Tm (.)Tj 0 -0.394.1411 1i,(˙)Tj /F9 1 Tg.

- Computation with polynomial nonlinearities.
- Advanced case splitting capabilities for the discovery of particular solution branches with desired properties.
- A visualization tool for the examination of the binary tree that results from multiple cases.
- Automatic generation of an existence and uniqueness theorem for the system.
- Algorithms for working with formal power series solutions of the system.

Applying RifSimp to the equations output by Dynaflex has the benefit of symbolically and automatically generating all special cases, through the RifSimp case-split options [9, 21]. In a full case analysis, some cases can be very complicated while others can be simple enough to be solved analytically. The canonical form generated by RifSimp is of low (0 or 1) di erential index [1, 10] which is suitable for the application of numerical solvers. An important option with RifSimp is the possibility of excluding special cases that are known to be not of interest. Thus if we know that a special case, say $m = 0$, is of no physical interest, then we can append the $m \neq 0$ to the input system [9, 21].

Application of the RifSimp package to multibody systems revealed that it has di culty handling large systems generated by Dynaflex. The symptoms

ton. Simple textbook problems in dynamics use choices of co-ordinate systems to produce simple systems of equations without constraints modelling the problems. But this method is usually not possible with complicated systems, which are automatically generated by packages such as Dynaflex and constraint equations can not be eliminated. In addition, the constraints introduce additional variables (essentially Lagrange multipliers) into the equations, representing the forces they exert.

2.1 Open Loop: Three-Dimensional Top

In this classic problem, the top is an axisymmetric body that can precess about a vertical (Z) axis, nutate about a rotated X axis, and spin about a body-fixed symmetry axis, see figure 1. The top is assumed to rotate without slipping on the ground; this is modelled by placing a spherical (ball-and-socket) joint at O. Dynaflex automatically generates co-ordinates using standard 3-1-3 Euler angles (, ,), which correspond to precession, nutation, and spin, respectively.

The top has a moment of inertia J about the symmetry axis, and A about an axis at O perpendicular to the symmetry axis. Two angles (the angle of the axis of symmetry to the axis), and specify the orientation of the axis of symmetry of the top, while specifies how a point on the top is moving relative to its axis of symmetry. There is a coordinate singularity when is equal to 0 or π, which RifSimp will automatically detect as part of its case analysis. Coordinate singularities are ubiquitous in automatically generated mechanical systems, and their automatic detection is an important problem.

The dynamic equations (1,2) generated by Dynaflex are, after changing from Dynaflex-generated symbols to more conventional ones, as follows. For details, we refer to the Dynaflex Users Guide [16].

$$M = \begin{bmatrix} (A\sin^2 \ + J\cos^2 \ ) & 0 & J\cos \\ 0 & A & 0A \end{bmatrix}$$

and

$$F = \begin{bmatrix} \sin(\;)\left[2(J-A)\cos(\;)\dot{\;} + J\dot{\;}\right]\dot{\;} \\ \sin(\;)\left[(A-J)\cos(\;)\dot{\;}^2 - J\ddot{\;} + mgl\right] \\ J\sin(\;)\dot{\;}\dot{\;} \end{bmatrix} \tag{4}$$

The fact that the top is an open-loop system is reflected in the fact that Dynaflex has generated 3 equations for 3 unknowns.

2.2 Two-Dimensional Slider Crank

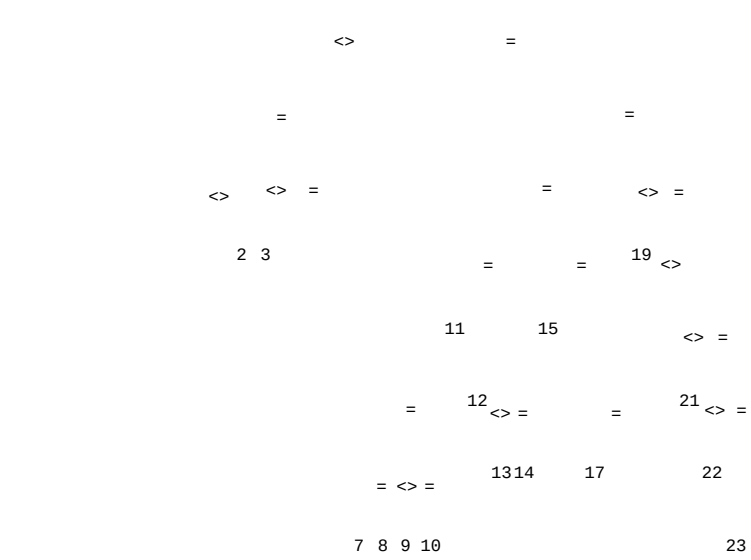
The slider crank is a simple example of a closed-loop system with

3 S •a s g R S Ca s S

Given that symbolic algorithms such as Dynaflex exist for automatically producing the equations modelling multi-body systems, it is natural to exploit the further simplification and transformation of such systems using symbolic methods. In this section we discuss the simplification of such systems using the Maple algorithm `casesplit` which is part of RifSimp.

The theory underlying the Reduced Involutive Form given in [12] applies to systems of analytic nonlinear PDE in dependent variables $_1, _2, \ldots _n$, which can be functions of several independent variables. While the general method applies to analytic systems, like most methods in computer algebra, the implemented algorithms apply to systems which are polynomial functions of their

3 .



u nv u v n h n 3

Definition 4.1 [Implicit Reduced In oluti e Form]: ≺ .
A $L = 0, N = 0$ $\gamma_1, ..., \gamma_k$ L $\gamma_1, ..., \gamma_k$
≺ (. .

Rewriting the system (15,16) in matrix form yields A, b, $[\vec{r}_1, \ldots, \vec{r}_k]^T$ in (19).

It remains to verify that D_t when reduced first with respect to L and then with respect to N yields zero for each $\in N$. First $D_t = {}_q \cdot + {}_t$ is unaltered by reduction with respect to L and then reduces to zero with respect to N (since it is already in N). Next $D_t({}_q \cdot + {}_t)$ is given in (16) and reduces to zero on simplification with respect to L

Our planned work includes other strategies for controlling the generation of large expressions, since there will always be a desire on the part of design engineers to increase the number of bodies that can be modelled. One strategy for large expression management (LEM) has been described in [4]. The key idea is that large expressions are not arbitrary collections of terms, but contain a structure. An analogy can be drawn with the situation in the study of matrices arising in engineering applications: they almost always have a 'structure' to them. For example, they are banded, or otherwise sparse. By recognizing structure, we can solve larger problems. Returning to symbolic manipulation, we can note that simplification routines in computer algebra can cause a loss of structure, usually with the result that larger expressions are generated. A very simple example is the apparent simplification of $(1+x)^9 - 1$, where a computer system will expand the bracket in order to cancel the 1 from the expansion with the 1 outside the bracket. Using the tools developed in [4] and now incorporated into Maple, we can preserve the structure inherent in engineering equations, such as those described here.

R s

1. . h n . . *C M O D E* *D* -*A E* . (1).
2. . u hb , . . n . *C A A C* - . n - (1 3).
3. . u. *C A M* , *J* . u b n n u . hn (1 3).
4. . . , . . , . . n n, bh . *P C*- *F M L* -*E M* , . b u n