Multibody System Dynamics. Vol. 14, pp. 61-80, 2005.

Dynamic Simulation of Multibody Systems Using a New State-Time Methodology

Kurt S. Anderson[*] and Mojtaba Oghbaei[†]
Department of Mechanical, Aerospace, and Nuclear Engineering
Rensselaer Polytechnic Institute, Troy, NY 12180-3590 USA

Multibody System Dynamics, Accepted for Publication

Abstract

This paper presents a new methodology demonstrating the feasibility and advantages of a *state-time* formulation for dynamic simulation of complex multibody systems which shows potential advantages for exploiting *massively parallel computing* resources. This formulation allows *time* to be discretized and parameterized so that it can be treated as a variable in a manner similar to the system state variables. As a consequence of such a *state-time* discretization scheme, the system of governing equations yields to a set of loosely coupled linear-quadratic algebraic equations that is well-suited in structure for some families of nonlinear algebraic equations solvers. The goal of this work is to develop efficient multibody dynamics algorithm that is extremely scalable and better able to fully exploit anticipated immensely parallel computing machines (tera flop, pecta flop and beyond) made available to it.

Keyword

Multibody dynamics, State-time formulation, Scalable algorithm, Parallel computing

Nomenclature

B	: Typical body B with its mass center B^*
$\vec{F}_{kA}$	: k^{th} applied force acting on the body B
$\vec{f}_{mC}$	: m^{th} unknown constraint force acting on the body B
$\vec{r}_{kA}$	: k^{th} position vector from B^* to application point of $\vec{F}_{kA}$
$\vec{r}_{B^*j_m}$	: m^{th}position vector from B^* to application point (joint) of $\vec{f}_{mC}$
$\vec{T}_l$	: l^{th} applied concentrated moment acting on the body B

[*]Associate Professor, anderk5@rpi.edu
[†]Doctoral Student, oghbam@rpi.edu

N	: Newtonian reference frame
$\vec{\vec{I}}^{B/B^*}$	: Inertia dyadic of body B with respect to B^*
${}^N\vec{\omega}^B$	: Absolute angular velocity of body B
${}^N\vec{\alpha}^B$	: Absolute angular acceleration of body B
ε_{ijk}	: The standard indicial cyclic permutation operator
$\vec{x}_B$	: Absolute displacement vector of B^*
${}^N\underline{\omega}^B_\times$	: Angular velocity cross product matrix
$\underline{C} = {}^N\underline{C}^B$	: Direction cosine matrix from the local frame B to N
$\psi_i(t)$ and $\varphi_j(t)$	: i^{th} and j^{th} members of families of C_1 and C_0 continuous shape functions, respectively

1 Introduction

Multibody systems (MBS) are defined as a collection of interconnected rigid and/or flexible bodies that can move relative to one another as permitted by their connecting joints. Physical systems that might be modelled as such are often called MBS and are pervasive in modern society. Consequently, multibody dynamics, as a discipline describing the dynamic behavior of such systems, plays a major role in the design and operation of such systems, and is enjoying broad, yet ever increasing diversity and frequency of application.

In general, the effective cost of simulation-based engineering consists of: *i)* The engineer's time to build the simulation model/code; *ii)* The cost associated with verifying and validating the model/simulation code; *iii)* The time required to run the necessary simulations (case studies); and *iv)* The cost associated with the subsequent analysis of the simulation results. To be effective, simulation-based engineering should significantly reduce the time and cost associated with design, construction, and testing of engineering systems. Such modelling and simulation capability, can thus greatly reduce the need for expensive prototyping and testing, as well as facilitate the control of the system. In many situations, fast and accurate modelling and simulation may be indispensable. Indeed, for many systems desired prototyping and testing may not even be a possibility and/or fast, accurate modelling for control, or operator/hardware in the loop testing may be required.

A variety of formulations and dynamic simulation algorithms have been developed by individuals that are sufficiently general to handle a wide range of MBS. However, the computational cost associated with many of these algorithms is considerable, thus limiting the extent to which they might be applied. This has driven the effort by many to develop more efficient (e.g. faster) formulations. These efforts first focused on serial processing alone with attention given to reducing the overall computational cost by clever (lower cost) manipulations and algebraic procedures. Later, as parallel computing resources became more available, researchers began to think about improving simulation speed and reducing turnaround time by producing parallel MBS algorithms. But these initial parallel attempts proved disappointing. For example, using a then state of the art parallel computing system and special parallel algorithms [1] [2] [3], the simulation of a controlled 30-second slew maneuver for a >160 flexible-body mode multibody model of a single solar array wing of the international space station was performed. The model of the space station had undesirably low fidelity, yet the simulation required on the order of 4 days of CPU time to complete. On a far simpler problem, Schwertassek [4] simulated the response of an off-road vehicle to a simple steering maneuver. The simulation used a parallel low order *Residual algorithm* on an eight-processor Transputer parallel processing system. The very modest 10 body, 18 joint, eight closed loop vehicle model required $\sim$

8.4 seconds for the simulation of a 5-second steering maneuver. Chung and Haug [5] realized only a factor of ~ 4 speedup in the simulation of an HMMWV using an 8 processor shared memory Alliant FX/8 and 'static' scheduling. A similar computational challenge and performance gains can be found in the field of biomechanics in [6]. The fact with these examples as well as the other most MBS applications is that the number of generalized coordinates used to describe the system is many orders of magnitude smaller than the number of temporal steps needed for a desired simulation. Thus these methods all show very limited *speedup* (reduced simulation turn-around) due to parallelization.

Examples of such undesirable simulation costs/time may be found in many branches of science and engineering including but not limited to dynamic systems, material modelling, flow modelling, combustion process, biomechanical systems, etc. This point was emphasized and repeated as an intrinsic problem at the SCaLeS workshop, held (7/03) in Washington D.C. [7] and in literature dealing with dynamic systems simulation. Three decades of MBS efforts and applications have demonstrated that there is and always will be considerably more formidable systems one wishes to model and analyze that can be treated using the most current tools and computing resources. Hence, it is essential that algorithms be developed and improved so that any gains in computing resources are fully exploited, enhanced and multiplied.

The major drawback of almost all contemporary multibody algorithms is that they are inherently sequential in time. Due to this characteristic, the focus of virtually all MBS formulations has been to reduce the cost per temporal integration step. In a parallel computing context, the effort has been to parallelize the governing dynamical equations *spatially* over the current temporal integration step. Parallel implementation of these formulations in most MBS applications is hobbled by sequential bottlenecks, and thus the use of additional processors beyond a very modest number will not increase the speed of the analysis in a significant way unless these sequential bottlenecks can be reduced. By comparison, parallelizing the simulation and all related analysis both *spatially* and *temporally* would result in a drastic increase in the number of coarse grain calculations that may be distributed over all the available processors.

Thus it is the prime objective of the research, proposed in this paper, to formulate a new multibody dynamic algorithm which may serve as an effective tool in the design and simulation of such systems. The algorithm presented in this paper provides the potential to yield significantly reduced simulation turnaround time achieved through its ability to better exploit massively parallel computing resources (e.g. *IBM/Sandia Blue Gene* [8] and beyond). The methodology is markedly distinct from the other traditional dynamics analysis and simulation methods, as well as prior attempts in using state-time approaches for dynamical systems, in terms of not being so limited to sequential (time stepping) solution, and extendability to complex systems.

2 *State-Time* vs. Traditional *State* Dynamic Formulations

Initially, the majority of multibody systems dynamics researchers tended to utilize either a *Newton-Euler* formulation, *Lagrange's* method or a combination thereof. However, due to some computational disadvantages associated with these schemes, dynamic analysis techniques based on what are broadly termed *velocity space projection methods* began to appear and have played a major, if not dominant, role in contemporary multibody dynamics work. Much effort has also been expended by investigators to develop algorithms with the aim of improving simulation turnaround. Such algorithms first strove for generality with little, if any, attention been paid to computational cost and speed. Then, once acceptable generality was achieved, greater emphasis was placed on improving simulation speed most often by reducing the underlying computational cost. Of these formulations, the so-called 'Order-n' or $O(n)$ algorithms have excelled for systems involving many generalized coordinates n, and relatively few additional geometric and motion constraints m. However, the development of such algorithms had largely been oriented towards application on purely sequential-architecture computer systems with relatively modest attention being directed to parallel computing issues. Jain [9] presents a survey of many of these $O(n)$ algorithms.

With the advent of parallel computing, researchers began to rethink the use of previously developed dynamic simulation procedures, and how the benefits of concurrent processing might be realized. This effort was pioneered by Kasahara *et. al.* [10] and followed by many investigators. Some of the highlights of these parallel multibody dynamics algorithms can be found in [11]- [19]. The focus of virtually all theses formulations has been to solve for the system state derivatives at the current time step in the most efficient parallel manner. The key point is that all calculations are only parallel within the given integration (time) step and the *simulation must sequentially march forward from the current time step to the next*, and then the parallel computations may be repeated.

These algorithms have resulted in a significant, but less than desired performance gains in simulation turnaround, which arise as a consequence of *Amdahl's Law* [20] and the fact that exchange of information between processors of a parallel processing system comes with a cost. Very roughly to put, Amdahl's Law states that the time required to run a computer calculation in parallel is asymptotically limited by the time required to perform the sequential portions of the calculation:

$$S_P = \frac{T_S}{T_P} = \frac{1}{f + \frac{(1-f)}{N_P} + \frac{N_P \cdot T_{Comm}}{T_S}}, \tag{1}$$

where S_P is the *speedup*, f is the fraction of the operations which must be performed sequentially in the given formulation; N_P is the number of processors used in the parallel calculation; T_{Comm} is the time required for an inter-processor communication with a single processor; T_P is the time required to perform the calculation in parallel; and T_S is the time required to perform the calculation sequentially. From Equation (1) it is clear that computation speedup can be adversely dominated by sequential bottlenecks and inter-processor communication costs.

Because many aspects of these dynamic formulations (particularly the *low order* formulations) have significant inherently sequential (causal) calculations, parallel implementation of these formulations are hobbled by sequential bottlenecks. Additionally, in most MBS applications to date the temporal domain of interest is far greater in scope than the spatial domain (e.g. # time steps $\gg$ # spatial variables). As there will generally be extremely many temporally integration steps needed to perform a desirable simulation, proceeding in this fashion will result in a fraction of sequential operations that are far greater than desired.

What is desired is to reformulate the equations of motion and temporal integrations such that *time* as well as the *spatial* variables is treated in an all encompassing manner. The proposed *state-time* dynamics formulation would allow the equations of motion to be parallelized temporally as well as spatially. This would have two advantages; First, the system of equations may now be coarse grain parallelized to a far greater degree. This will allow an increased number of processors N_P, which can be effectively utilized. Secondly, this would significantly reduce the fraction of sequential operations and thus increase speedup (reduced turnaround).

The idea of discretizing equations in time domain, generally known as space-time formulations, is not a new concept. Initial investigations on space-time finite element formulation were proposed by

Argyris *et al.* [21] and Fried [22] in the late 60s. These works were not initially paid much attention because of the enormous demands for the size of main storage needed. However, the advancement of MIMD-parallel computers, and modern iterative solvers with multilevel strategy, initiated a revival for such approaches. These advances have made it possible to utilize some attractive features of simultaneous space-time discretization. Dynamics community have also considered this type of approach in regard to various applications such as structural dynamics, nonlinear vibrations, inverse dynamics of flexible robots, gate analysis, and optimal control problem for converting problems from infinite dimension to finite dimension. Some of the related works can be found in [23]- [29]. These prior time-parameterization methods often produced highly nonlinear and coupled sets of equations, thus limiting their tractability with complex systems. To the authors' knowledge none of these prior time-parameterization efforts have been performed with the intent of developing a general multibody formulation which may more fully exploit future massively parallel computing resources.

3 The New *State-Time* Methodology

Briefly, formulating the problem in the proposed *state-time* method is achieved by writing the variational form of the governing equations and then applying the *Galerkin* approximation using polynomial interpolating functions. It should be noted that the use of collocation methods for time discretization is another possibility. However, in these methods the approximating function (usually a polynomial) is required to satisfy the given equations at some collocation points and the error control criteria can only be applied on the set of function approximated values on the boundaries of the integration steps. But there are some advantages of using the weak form of the governing equations of motion in the proposed scheme which are: first, it enables the use of lower order shape functions to approximate state variables; second, it allows dealing with a priori known impulsive forces in a more convenient way; and third, it provides the means for stabilizing the error over the entire time step in a minimum energy sense. As indicated above, it is desirable to formulate the problem in such a way that it is not limited to a largely sequential temporal marching scheme. This can be accomplished by treating the *time* appearing within the equations of motion as a variable in much the same manner as having been done on the spatial coordinates by the finite element community.

In this section, formal derivation of *state-time* equations is given for a unit temporal element. Mapping these equations into the physical temporal domain where the initial and final time could be arbitrarily t_0 and t_f, as well as using them in an assembly operation are trivial tasks that are not referred to in this article. For the purpose of generating an associated system of equations which provide the least coupling and lowest level of nonlinearity, as well as the greatest degree of coarse grain parallelism, a *Newton-Euler* approach is taken. As a result of this approach, all orientation, position and velocity variables, as well as kinematic constraint forces will be parameterized in time producing an associated set of *state-time* variables. These variables appear in the associated *state-time* equations, and are the unknowns which are to be solved for. Below, it is shown how to produce the *state-time* representation of each of the *Euler's* equation, the generalized form of *Newton's* 2^{nd} *Law*, *Poisson's kinematical equations* and geometric constraint equations for a typical body within a multibody system.

3.1 *State-Time* Consideration of the *Euler's Equation*

Euler's equation for the rotational motion of an arbitrary body B can be written as

$$\sum \vec{M}^{B/B^*} = \sum_k \vec{r}_{kA} \times \vec{F}_{kA} + \sum_m \vec{r}_{B^* j_m} \times \vec{f}_{mC} + \sum_l \vec{T}_l = \vec{\vec{I}}^{B/B^*} \cdot {}^N\vec{\alpha}^B + {}^N\vec{\omega}^B \times \vec{\vec{I}}^{B/B^*} \cdot {}^N\vec{\omega}^B \tag{2}$$

or for simplicity, in indicial form

$$\varepsilon_{ijh}\, r_{kj}\, C_{ph}\, F_{kp} + \varepsilon_{ijs}\, r_{mj}\, C_{nk}\, f_{mn} + C_{qi}\, T_q = I_{ij}\, \alpha_j + \varepsilon_{ijt}\, \omega_j\, I_{tl}\, \omega_l \tag{3}$$

In the above equations let us consider

$$\alpha_i = \dot{\omega}_i = \ddot{q}_i \tag{4}$$

where q exists mathematically, but not necessarily physically. Applying weighted residual method on equation (3) and then integration by part results in

$$\begin{aligned} &\int_0^1 \{\varepsilon_{ijh}\, r_{kj}\, C_{ph}\, F_{kp} + \varepsilon_{ijs}\, r_{mj}\, C_{nk} f_{mn} + C_{qi}\, T_q\}\, \tilde{E} dt - \\ &\qquad I_{ij} \left(\dot{q}_j \tilde{E}\Big|_0^1 - \int_0^1 \dot{q}_j \dot{\tilde{E}} dt \right) - \varepsilon_{ijt} I_{tl} \int_0^1 \dot{q}_j\, \dot{q}_l\, \tilde{E} dt = 0 \end{aligned} \tag{5}$$

Note that capital letters with *tilde* sign denote to the weighting functions and the barred quantities refer to the *state-time* variables. By defining the following parameterized relations in time

$$\left. \begin{array}{lll} q_j = \overline{q}_{ju} \psi_u(t) & ; & C_{ph} = \overline{C}_{ph_x} \psi_x(t) \\ f_{mn} = \overline{f}_{mn_t} \varphi_t(t) & ; & \tilde{E} = \overline{E}_r \psi_r(t) \end{array} \right\} \tag{6}$$

the Galerkin approximation of equation (5) after simplification may be written as

$$\begin{aligned} &\overline{C}_{ph_x} \cdot \left\{ \varepsilon_{ijh}\, r_{kj}\, F_{kp} \int_0^1 \psi_x(t) \cdot \psi_r(t) dt \right\} + \overline{C}_{qi_y} \cdot \left\{ T_q \int_0^1 \psi_y(t) \cdot \psi_r(t) dt \right\} + \\ &\qquad \overline{C}_{nsw} \cdot \overline{f}_{mn_t} \cdot \left\{ \varepsilon_{ijs}\, r_{mj} \int_0^1 \psi_w(t) \cdot \varphi_t(t) \cdot \psi_r(t) dt \right\} - \\ &\quad I_{ij} \left\{ \left[\overline{q}_{j_u} \dot{\psi}_u(1) \cdot \psi_r(1) - \omega_j(0) \cdot \psi_r(0) \right] - \overline{q}_{j_u} \int_0^1 \dot{\psi}_u(t) \cdot \dot{\psi}_r(t) dt \right\} - \\ &\qquad \overline{q}_{j_u} \cdot \overline{q}_{lv} \cdot \left\{ \varepsilon_{ijt}\, I_{tl} \int_0^1 \dot{\psi}_u(t) \cdot \dot{\psi}_v(t) \cdot \psi_r(t) dt \right\} = 0 \end{aligned} \tag{7}$$

This is the *state-time* representation of the *Euler's* equation which in general constitutes $3 \cdot k \cdot n_e$ nonlinear algebraic equations and $9 \cdot k \cdot n_e + 3 \cdot m \cdot (p \cdot n_e + 1) + 3 \cdot k \cdot n_e$ unknown *state-time* variables. n_e represents the number of temporal elements and m is the number of geometric constraints. Also, k and p denote to the order of the shape functions associated with spatial and force variables respectively, and in general are different so that the *Babuška-Brezzi condition* [30] is satisfied.

3.2 *State-Time* Consideration of the Generalization of *Newton's* 2^{nd} *Law*

In the *Newton's* 2^{nd} *Law* as stated below, it is assumed that all active forces included in $\vec{F}$ are known. However, in many cases some of these constituent forces might be functions of the state of the system, e.g. the spring force or viscous force. In such situations, we can still apply the same discretization scheme on these state-dependent forces.

$$\vec{F}^B + \sum_m \vec{f}_{mC} = m_B \, \ddot{\vec{x}}_B \tag{8}$$

where $\vec{F}^B = \sum_k \vec{F}^B_{kA}$ is the resultant of applied (state dependent) forces and $\vec{f}_{mC}$ are the unknown constraint forces. For simplicity, we neglect the superscript B in the subsequent equations. Following a similar scheme as described above, the Galerkin approximation of equation (8), results in

$$\begin{aligned} &\left\{\overline{x}_{ij}\dot{\psi}_j(1)\cdot\dot{\psi}_r(1) - \dot{x}_i(0)\cdot\dot{\psi}_r(0)\right\} - \overline{x}_{ij}\int_0^1 \dot{\psi}_j(t)\cdot\dot{\psi}_r(t)dt - \\ &\frac{1}{m}\left\{F_i\int_0^1 \psi_r(t)dt + \sum_m\left[\overline{f}_{mk}\cdot\int_0^1 \varphi_k(t)\cdot\psi_k(t)dt\right]\right\} = 0 \qquad i = 1,2,3 \end{aligned} \tag{9}$$

Equations (9) are the *state-time* representation of *Newton's* 2^{nd} *Law* which constitutes $3\cdot k\cdot n_e$ linear algebraic equations and introduces $3\cdot k\cdot n_e$ additional unknown *state-time* variables pertaining to translational DOF of the body B.

3.3 *State-Time* Consideration of the *Poisson's Kinematical Equations*

In the spatial dynamics of MBS, the components may undergo large translational and rotational motions. To define the configuration of each body with respect to neighboring reference frame, it is convenient to assign a reference frame to each body.

One of the most widely used parameterization in describing reference frame orientations are the use of three independent orientation angles (of which *Euler's angles* are a subset). A three parameter set is attractive since it has the same number of parameters as there are rotational DOF for a body. A disadvantage of this representation is that no three-parameter set can be both global and nonsingular [31]. Additionally, this type of parameterization is not considered in the context of *state-time* formulation because the resulting equations would become highly coupled in an undesirable nonlinear fashion.

To be both nonsingular and global, more than three parameters are required to set up the basis vector transformation matrix. *Euler's parameters* are a set of four parameters that are well suited for computation and have been used in several general-purpose multibody programs since they do not suffer from singularity. Unfortunately, *the kinematical differential equations* associated with Euler's parameters are individually quadratic in the system variables. This results in final *state-time* equations that would be cubic in *state-time* variables. Hence, these parameters also yield a level of coupling and nonlinearity in the subsequent calculations that is not desirable from the viewpoint of this research.

An alternate way of describing the three dimensional rotations in MBS dynamics is the direct use of *direction cosines* as rotation parameters. For all the above mentioned reasons, we have realized that if the equations of motion are written intelligently, this form of rotation representation results in the set of algebraic equations that are quadratic in *state-time* variables, with the lowest level of coupling. If these direction cosines are used directly as state variables, then the kinematical differential equations, which relate the time derivative of the transformation matrix to the angular velocity vector of the body are *Poisson's equations* given by

$$^{N}\underline{\dot{C}}^{B} = {}^{N}\underline{C}^{B} \cdot \left[{}^{N}\underline{\omega}^{B}_{\times}\right]_{B} \tag{10}$$

or in indicial form

$$\dot{C}_{il} = -\varepsilon_{mlk}\,\omega_k\,C_{im} \tag{11}$$

Applying the weighted residual and substituting the approximation relations yields

$$\overline{C}_{iln} \cdot \int_0^1 \dot{\psi}_n(t)\psi_r(t)dt + \overline{C}_{imj} \cdot \left\{\varepsilon_{mlk}\,\overline{q}_{kp} \int_0^1 \dot{\psi}_p(t) \cdot \psi_j(t) \cdot \psi_r(t)dt\right\} = 0 \tag{12}$$

Equations (12) are the *state-time* representation of *Poisson's* equations which constitutes $9 \cdot k \cdot n_e$ quadratic algebraic equations and there is no additional unknown *state-time* variable introduced by them.

3.4 ***State-Time*** Consideration of the Geometric Constraint Equations

Consider the body B with m neighbors as illustrated in Figure 1

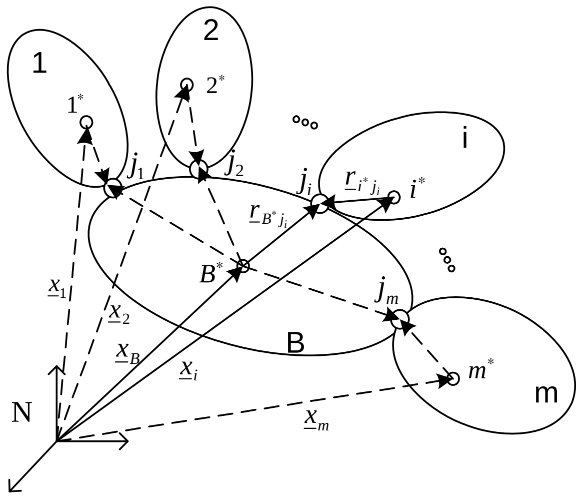


Figure 1: A multibody system in tree structure

For each joint J_i we can write the algebraic constraint relationship

$$\underline{x}_B + {}^{N}\underline{C}^{B}\,\underline{r}_{B^*j_i} = \underline{x}_i + {}^{N}\underline{C}^{i}\,\underline{r}_{i^*j_i} \xrightarrow{\text{vel. level}} \underline{\dot{x}}_B + {}^{N}\underline{\dot{C}}^{B}\,\underline{r}_{B^*j_i} = \underline{\dot{x}}_i + {}^{N}\underline{\dot{C}}^{i}\,\underline{r}_{i^*j_i} \tag{13}$$

Applying the weighted residual method on each of these equations gives

$$\int_0^1 \left\{\underline{\dot{x}}_B + {}^{N}\underline{\dot{C}}^{B}\,\underline{r}_{B^*j_i}\right\}\tilde{G}_i dt = \int_0^1 \left\{\underline{\dot{x}}_i + {}^{N}\underline{\dot{C}}^{i}\,\underline{r}_{i^*j_i}\right\}\tilde{G}_i dt \qquad i = 1, 2, ..., m \tag{14}$$

The Galerkin approximation of equations (14) after parameterization in time may be written as

$$\overline{x}_{Btj} \int_0^1 \dot{\psi}_j(t) \cdot \varphi_z(t)dt + {}^N\overline{C}^B_{tkm} \int_0^1 r_{B^* j_{i\,k}} \dot{\psi}_m(t) \cdot \varphi_z(t)dt -$$
$$\overline{x}_{its} \int_0^1 \dot{\psi}_s(t) \cdot \varphi_z(t)dt + {}^N\overline{C}^i_{tln} \int_0^1 r_{i^* j_{i\,l}} \dot{\psi}_n(t) \cdot \varphi_z(t)dt = 0 \qquad i = 1, 2, ..., m \tag{15}$$

Equations (15) are the *state-time* representation of geometric constraint equations. Before counting the number of new equations and independent unknowns due to these equations, let us first look at the correspondent *Euler's* equation, generalization of the *Newton's* 2^{nd} law, and *Poisson's* equation associated with each of the attached bodies. In a similar manner as elaborated for body B, we can find the number of new equations and unknowns introduced by each of these equations for these bodies. At the end, after counting the number of independent equations and unknowns pertaining to the *state-time* representation of the common geometric constraint equations, the number of algebraic *state-time* equations and unknowns for the entire system are determined to be

$$15\, k\, n_e\, (m+1) + 3\, m\, (p\, n_e + 1) \tag{16}$$

The resulting nonlinear algebraic system of equations, though large in dimension, is sparse and exhibiting principally quadratic nonlinearity, enabling parallel iterative solution techniques to be used effectively. In the following section, the *state-time* scheme is implemented on a planar double pendulum system and the associated results have been illustrated.

4 Example Application - Planar Double Pendulum

The double pendulum, as depicted in Figure 2, is a simple mechanical system that exhibits complex dynamic behavior. In this section, we have presented the results of a 5 second simulation based on the *state-time* formulation and the MATLAB ODE45 solver, which are superimposed on the same plots for better comparison.

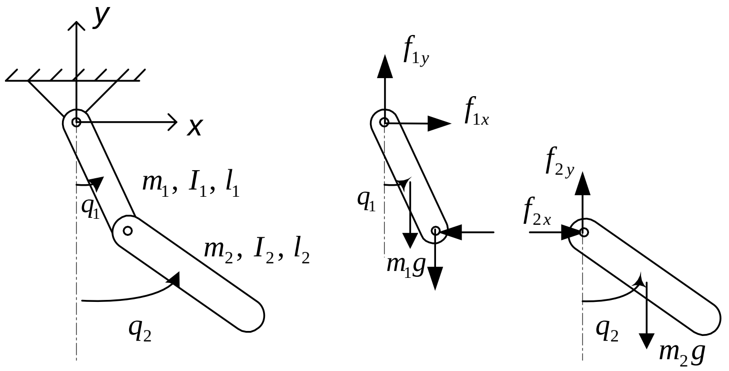


Figure 2: Planar Double Pendulum

Dynamical equations using the Newton-Euler's formulation are consisted of seven equations and seven unknowns per each body. In an effort to preserve space, the governing equations are presented only for representative body 1, and are given in Equations (17). It is noted that even though C_q and S_q or x and y can be simply found from q, for the reasons explained in section 3 they are treated as independent variables. The following system of equations is a prototype of a *mixed problem* [30], where both constraint forces f_x and f_y as algebraic variables and spatial coordinates q, x, y, C_q and S_q as differentiated variables are present as unknowns.

$$\begin{array}{rcl} Euler's\ equation & : & I_1\,\ddot{q}_1 + (f_{1x}+f_{2x})\dfrac{l_1}{2}\,C_{q_1} + (f_{1y}+f_{2y})\dfrac{l_1}{2}\,S_{q_1} = 0 \\ Newton's\ 2^{nd}\ Law & : & \left\{ \begin{array}{l} m_1\ddot{x}_1 = f_{1x} - f_{2x} \\ m_1\ddot{y}_1 = f_{1y} - f_{2y} - m_1 g \end{array} \right. \\ Poisson's\ equation & : & \left\{ \begin{array}{ll} \dot{C}_{q_1} = -\dot{q}_1\,S_{q_1} & (\,C_{q_1} = cos(q_1)\,) \\ \dot{S}_{q_1} = \dot{q}_1\,C_{q_1} & (\,S_{q_1} = sin(q_1)\,) \end{array} \right. \\ kinematic\ constraints & : & \left\{ \begin{array}{l} \dot{x}_1 = \frac{l_1}{2}\,\dot{q}_1\,C_{q_1} \\ \dot{y}_1 = \frac{l_1}{2}\,\dot{q}_1\,S_{q_1} \end{array} \right. \end{array} \tag{17}$$

If the linear and quadratic Lagrange shape functions ($\vec{\varphi}(t)$ and $\vec{\psi}(t)$) are used to interpolate constraint forces and spatial variables, respectively, and then the *state-time* representation of the system of Equations (17). This results in a set of 14 quadratic-linear algebraic equations and 14 *state-time* variables per each temporal element.

State-Time representation of Euler's Equations

$$\begin{gathered} \int_0^1 \left[I_1\,\ddot{q}_1 + (f_{1x}+f_{2x})\frac{l_1}{2}\,C_{q_1} + (f_{1y}+f_{2y})\frac{l_1}{2}\,S_{q_1} \right] \tilde{Q}(t)dt = 0 \qquad \rightarrow \\ I_1 \left[\dot{q}_1(t)\,\tilde{Q}(t)\Big|_0^1 - \int_0^1 \dot{q}_1(t)\,\dot{\tilde{Q}}(t)dt \right] + \frac{l_1}{2}\int_0^1 \left[(f_{1x}+f_{2x})\,C_{q_1} + (f_{1y}+f_{2y})\,S_{q_1} \right] \tilde{Q}(t)dt = 0 \\ q_1(t) = \bar{q}_{1_\alpha}\,\psi_\alpha(t)\ \ ;\ \ C_{q_1}(t) = \bar{C}_{q_{1\gamma}}\,\psi_\gamma(t)\ \ ;\ \ f_{1x}(t) = \bar{f}_{1x_\eta}\,\psi_\eta(t)\ \ ;\ \ \tilde{Q}(t) = \tilde{Q}_\beta\,\psi_\beta(t) \\ \Rightarrow I_1 \left[\dot{q}_1(t)\,\psi_\beta|_0^1 - \bar{q}_{1_\alpha} \int_0^1 \dot{\psi}_\alpha\,\dot{\psi}_\beta dt \right] + \frac{l_1}{2}\ \left[(\bar{f}_{1x_\eta} + \bar{f}_{2x_\eta})\,\bar{C}_{q_{1\gamma}} + (\bar{f}_{1y_\eta} + \bar{f}_{2y_\eta})\,\bar{S}_{q_{1\gamma}} \right] \int_0^1 \psi_\eta\,\psi_\gamma\,\psi_\beta dt = 0 \\ \alpha\,,\gamma = 1,2,3 \quad ; \quad \eta\,,\beta = 1,2 \end{gathered} \tag{18}$$

State-Time representation of Newton's 2^{nd} law in the x-direction

$$\begin{gathered} \int_0^1 \left[m_1\,\ddot{x}_1 - (f_{1x} - f_{2x}) \right] \tilde{X}(t)dt = 0 \qquad \rightarrow \\ x_1(t) = \bar{x}_{1_\delta}\,\psi_\delta(t) \quad ; \quad f_{1x}(t) = \bar{f}_{1x_\eta}\,\psi_\eta(t) \quad ; \quad \tilde{X}(t) = \tilde{X}_\beta\,\psi_\beta(t) \\ \Rightarrow \quad m_1 \left[\dot{x}_1(t)\,\psi_\beta|_0^1 - \bar{x}_{1_\delta} \int_0^1 \dot{\psi}_\delta\,\dot{\psi}_\beta dt \right] - (\bar{f}_{1x_\eta} - \bar{f}_{2x_\eta}) \int_0^1 \varphi_\eta\,\psi_\beta dt = 0 \\ \delta = 1,2,3 \quad ; \quad \eta\,,\beta = 1,2 \end{gathered} \tag{19}$$

and similarly in y direction

$$\begin{gathered} m_1 \left[\dot{y}_1(t)\,\psi_\beta|_0^1 - \bar{y}_{1_\delta} \int_0^1 \dot{\psi}_\delta\,\dot{\psi}_\beta dt \right] - (\bar{f}_{1y_\eta} - \bar{f}_{2y_\eta}) \int_0^1 \varphi_\eta\,\psi_\beta dt + m_1 g \int_0^1 \psi_\beta dt = 0 \\ \delta = 1,2,3 \quad ; \quad \eta\,,\beta = 1,2 \end{gathered} \tag{20}$$

State-Time representation of the first Poisson's equation

$$\int_0^1 \left[\dot{C}_{q1} + \dot{q}_1\, S_{q1}\right] \tilde{P}(t)dt = 0 \;\Rightarrow\; \bar{C}_{q1_\gamma} \int_0^1 \dot{\psi}_\gamma dt + \bar{q}_{1_\alpha}\, S_{1q_\gamma} \int_0^1 \dot{\psi}_\alpha\, \psi_\gamma\, \psi_\beta dt = 0$$
$$C_{q1}(t) = \bar{C}_{q1_\gamma}\, \psi_\gamma(t) \quad ; \quad \tilde{P}(t) = \tilde{P}_\beta\, \psi_\beta(t)$$
$$\alpha,\, \gamma = 1,2,3 \quad ; \quad \beta = 1,2 \tag{21}$$

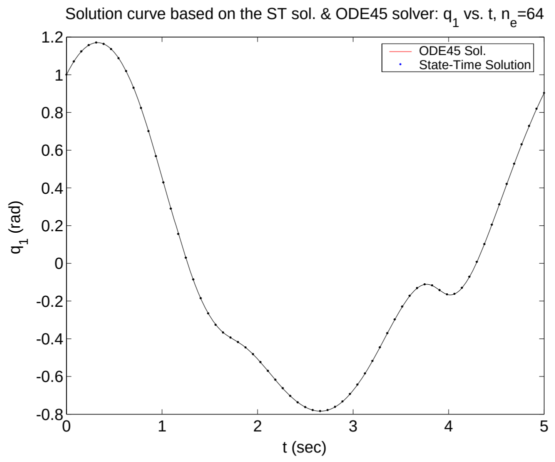
and for the second Poisson's equation

$$\bar{S}_{q_\gamma} \int_0^1 \dot{\psi}_\gamma dt - \bar{q}_\alpha\, C_{q_\gamma} \int_0^1 \dot{\psi}_\alpha\, \psi_\gamma\, \psi_\beta dt = 0 \tag{22}$$

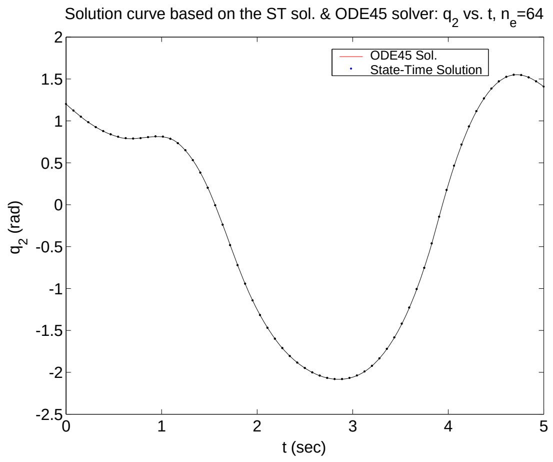
Finally, the state-time representation of the kinematic constraints may be similarly written as

$$\int_0^1 \left[x_\delta \dot{\psi}_\delta dt - l\, \bar{q}_\alpha\, C_{q_\gamma}\, \dot{\psi}_\alpha\, \psi_\gamma\right] \varphi_\eta dt = 0 \tag{23}$$
$$\int_0^1 \left[y_\delta \dot{\psi}_\delta dt - l\, \bar{q}_\alpha\, S_{q_\gamma}\, \dot{\psi}_\alpha\, \psi_\gamma\right] \varphi_\eta dt = 0 \tag{24}$$

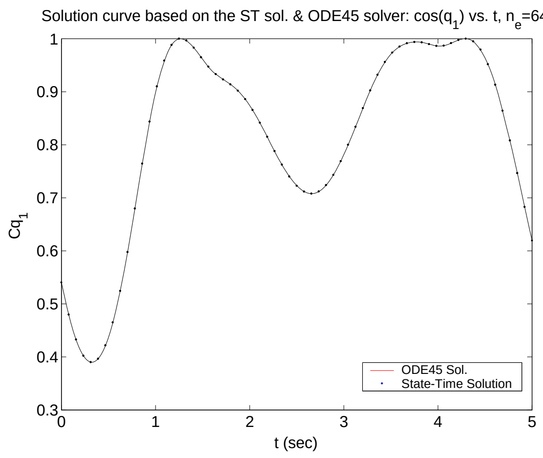
The parameters used for this simulation are $m_1 = 1$ kg, $l_1 = 0.8$ m, $\theta_{10} = 1$ rad, $\omega_{10} = 1$ rad/sec, $m_2 = 0.5$ kg, $l_2 = 1$ m, $\theta_{20} = 1.2$ rad, $\omega_{20} = -1$ rad/sec, and $g = 2$ m/sec^2. The plots given in Figure 3 show a 5 sec simulation of this system using 64 temporal elements. using linear/quadratic Lagrange shape functions to interpolate constraint forces and the spatial variables, respectively. Since for this system $m = 4$ (considering both pendulums), $k = 2, p = 1, n_e = 64$, the *state-time* representation of the entire system of equations results in a system of $28 + 24 \cdot 63 = 1540$ quadratic-linear algebraic equations and 1540 *state-time* variables.



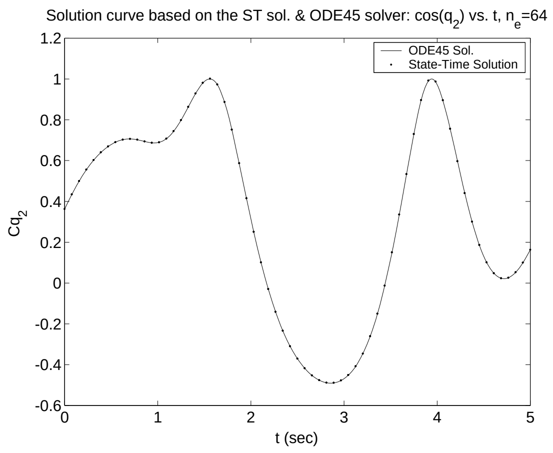
(a) Angular displacement, q_1 vs. t



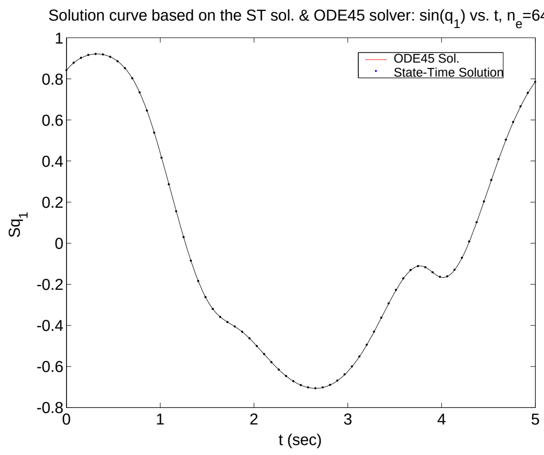
(b) Angular displacement, q_2 vs. t



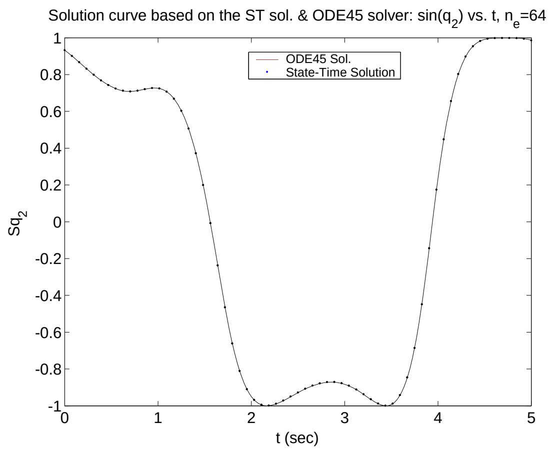
(c) C_{q1} vs. t



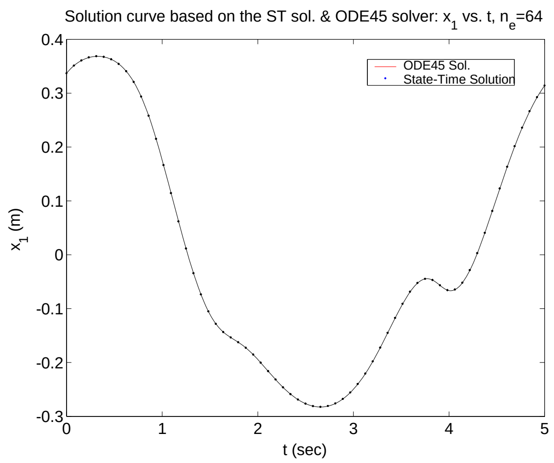
(d) C_{q2} vs. t



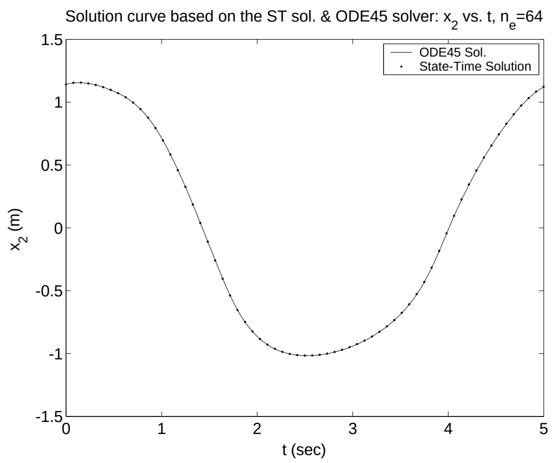
(e) S_{q1} vs. t



(f) S_{q2} vs. t

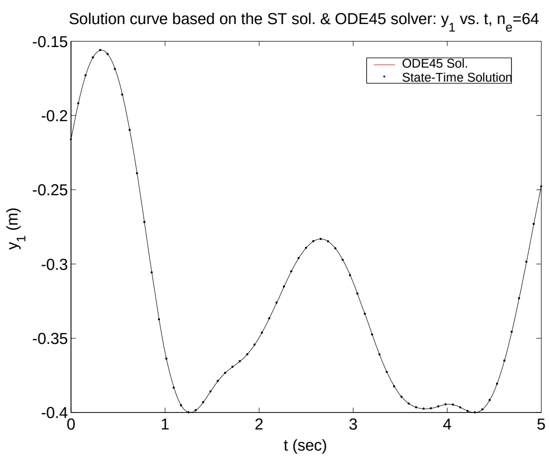


(g) Position of the mass center, x_1 vs. t

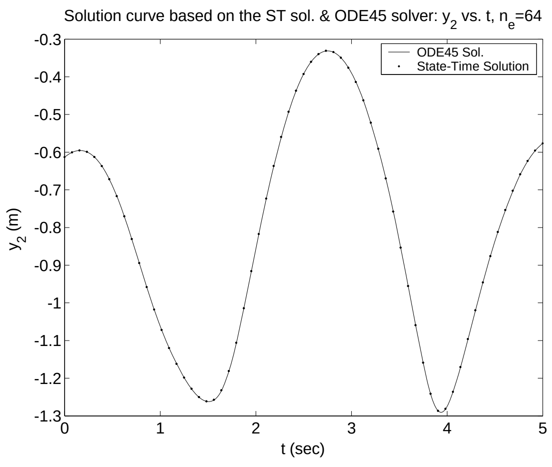


(h) Position of the mass center, x_2 vs. t

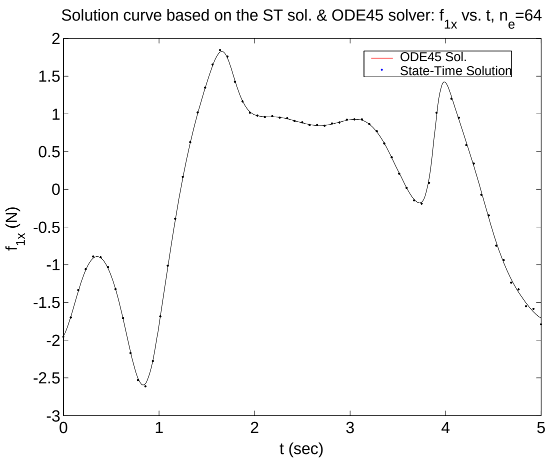
Figure 4 shows the sparse low-bandwidth structure of the *tangent matrix* A associated with the



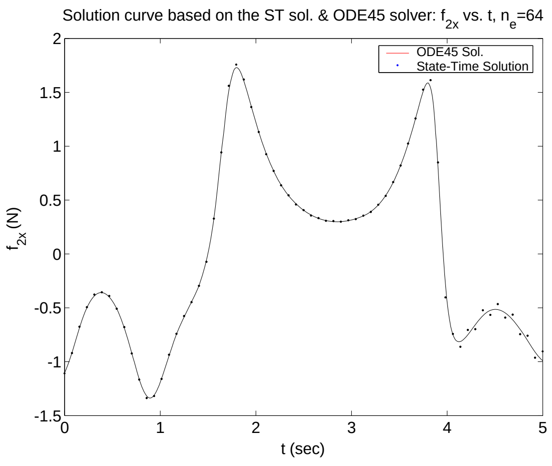
(i) Position of the mass center, y_1 vs. t



(j) Position of the mass center, y_2 vs. t



(k) Constraint force, f_{1x} vs. t

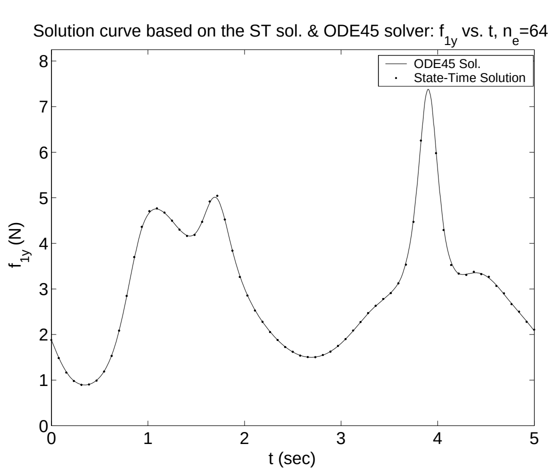


(l) Constraint force, f_{2x} vs. t

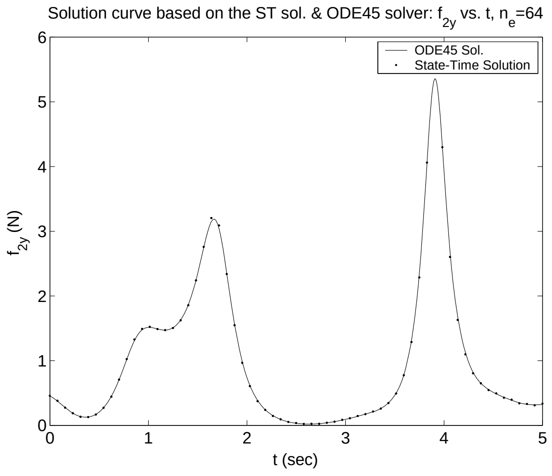
Newton-method solution of the set of nonlinear algebraic *state-time* equations for the above example. These equations have resulted in a highly desirable structure, namely a hybrid linear-quadratic system of loosely coupled equations in spatial and force variables with only nearest neighbor coupling in the temporal elements, enabling parallel iterative solution techniques to be used effectively.

Figure 5 illustrates how the accuracy of solution is improved as the number of temporal elements are increased within the time period considered. A comment needs to be made here on the order of accuracy of the method. Since we have selected quadratic shape functions to interpolate the spatial variables and linear approximation for the constraint forces, it is expected that a super-linear order of accuracy is acheived for the computation given above. This result is manifested in figure 5. Similarly, super-quadratic order of accuracy is anticipated if a combination of cubic and quadratic interpolating functions are used for the spatial variables and constraint forces, respectively.

Table 1 shows the numerical values of the results given in Figure 5 for more convenient reference. Here, we mean the *error* as the L_2 norm of the area encapsulated between the solution curves from the *state-time* scheme and ODE45 solver.



(m) Constraint force, f_{1y} vs. t



(n) Constraint force, f_{2y} vs. t

Figure 3: Simulation results of the planar double pendulum

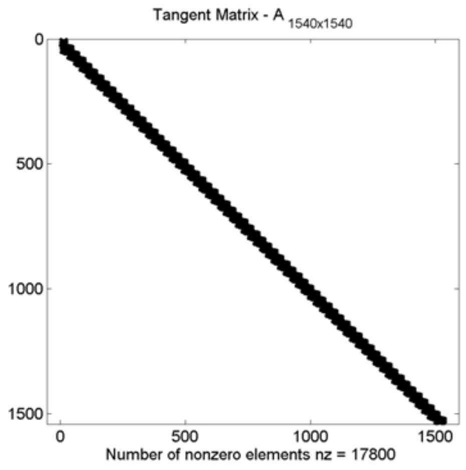


Figure 4: Sparse structure of the *tangent matrix*

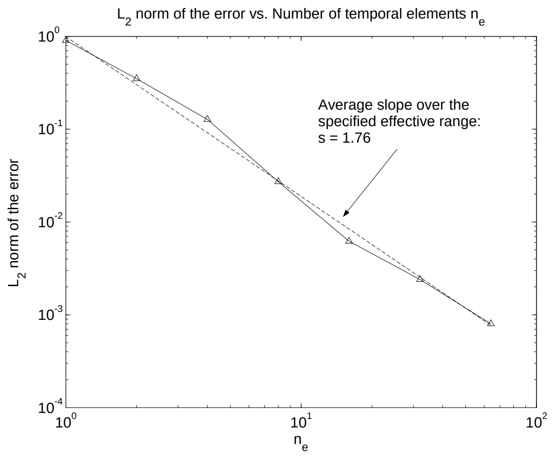


Figure 5: Accuracy improvement vs. number of temporal elements

Table 1: Accuracy improvement vs. number of temporal elements

Number of temporal elements (n)	1	2	4	8	16	32	64
L2 norm of the error	0.9038	0.3502	0.1273	0.0272	0.0062	0.0024	0.0008

5 Selection of Initial Guess

When running the implementation cases, a set of randomized initial iterates was also used to check for the robustness of the method. Some of these quantities were selected without considering some governing relations present in the system, for instance the orthonormality property of direction cosine matrix. As a rule, the more accurate the initial guess is, the less number of iterations that will be required for the Newton's scheme to converge. The number of required iterations for convergence of the method can be expected to keep increasing if a finer element size and longer simulation are desired. Having the initial position- and velocity-level data and the governing constraint equations are advantageous and can greatly help to achieve a high quality initial guess for the other unknown *state-time* variables. Thus it is critically important that such a task is done at first in order to run the *state-time* algorithm more efficiently.

Stochastic optimization algorithms like *Genetic Algorithm* (GA) have proven to be effective in searching enormous solution spaces efficiently. They can be used as a tool in quickly and efficiently exploring the likely candidates in the solution space to find a high quality initial guess for the *state-time* algorithm. As the goal of this work is on parallelism, an efficient parallelizable domain approximation strategy using genetic crossover [32] may be adopted in performing such a task as a preprocessing step for the *state-time* simulation. It is already shown that such techniques, when adopted on parallel machines, are extremely rapid without any need for decomposing the equations or calculating differentiations.

6 Conclusion

In summary, a traditional dynamic simulation treatment involving n generalized coordinates and m independent constraints will result in a system of $O(n+m)$ differential algebraic equations. These equations must be repeatedly solved at each time step for the system state derivatives, which are then temporally integrated. The results from current integration step are then used to update the system state and the process repeats as the simulation marches sequentially through temporal integration steps. These time steps often reflect the finest governing time scales within the system at the instant under consideration, and as such, reduce simulation speed because all state variables are integrated as dictated by these scales. Such drawbacks cause unavoidable computational cost for the entire simulation as well as the major issue of lack of scalability in time. The proposed methodology demonstrates great promise for improved speedup which can be achieved by circumventing the shortcomings of the recent algorithms. For instance, the dynamic behavior of a real life application, namely a 6 degree-of-freedom laser-powered lightcraft [33], is currently being investigated using the proposed method. Due to high spin and precession rates which are necessary to preserve flight stability, more than 10^5 temporal elements or time steps are required to properly model the lightcraft dynamics. The simulation turnaround that one can achieve through parallel state dynamics algorithms is theoretically $T_{simulation} = N_{steps} \cdot O(\log_2 n) = 10^5\, O(\log_2 6) \sim O(10^5)$. By comparison, if one has sufficiently many processors (with ideal communication) available, then the state-time approach would offer the possibility of reducing simulation turnaround time into $T_{simulation} = O(\log_2 N_{elements}) + O(\log_2 n') = O(\log_2 10^5) + O(\log_2 26) \sim O(20)$. Additionally, what makes this work distinct from the other space-time formulations that have been applied in a wide range of applications, is that all those implementations tend to result in a dense, highly nonlinear and heavily coupled structure that makes solution difficult, if not intractable. The *state-time* formulation presented here, which shows the capability of scaling up both spatially and temporally, results in a large dimension, lightly coupled, linear-quadratic system of equations offering a drastic increase in the number of coarse grain calculations that can be distributed over all the available processors of the parallel computing machine.

Acknowledgement

Support for this work has been provided by National Science Foundation (NSF) under the Award No. CMS-0219734 and is gratefully appreciated.

References

[1] S. S. Lee, "Symbolic Generation of Equations of Motion for Dynamics/Control Simulation of Large Flexible Multibody Space Systems" PhD thesis, University of California, Los Angeles, 1988

[2] K. S. Anderson, "Recursive Derivation of Explicit Equations of Motion for Efficient Dynamic/Control Simulation of Large Multibody Systems" PhD thesis, Stanford University, 1990

[3] R. Gluck, "A Custom Architecture Parallel Processing System for Space Station" Technical Report EML-003, TRW Space and Technology Group, Redondo Beach, CA, May 1989

[4] R. Schwertassek, "Reduction of Multibody Simulation Time by Appropriate Formulation of the Dynamics System Equations" In M. F. O. Pereira and J. A. C. Ambrosio, editors, Computer-Aided Analysis of Rigid and Flexible Mechanical Systems, NATO ASI, pages 447-482, Kluwer Academic Press, 1994

[5] S. Chung and E. J. Haug, "Real-time Simulation of Multibody Dynamics on Shared Memory Multiprocessors" J of Dynamic Systems, Measurement and Control, 115:628-637, Dec. 1993

[6] F. C. Anderson, J. M. Ziegler, and M. G. Pandy, "Numerical Computation of Optimal Controls for Large-Scale Musculoskeletal Systems" Advances in Bioengineering, 26:519-522, 1993

[7] "Science Case for Large-scale Simulation", June 24 & 25, 2003, Washington D.C., U.S.A, http://www.pnl.gov/scales/

[8] F. Allen, G. Almasi, et al., "Blue gene: A vision for protein science using a pectaflop supercomputer", IBM Systems Journal, 40(2):310-327, 2001

[9] A. Jain, "Unified formulation of dynamics for serial rigid multibody systems" Journal of Guidance, Control, and Dynamics, 14(3):531-542, May-Jun. 1991

[10] H. Kasahara, H. Fujii, and M. Iwata, "Parallel processing of robot motion simulation" In Proceedings IFAC 10^{th} World Conference, 1987

[11] D.S. Bae, J. G. Kuhl, and E. J. Haug, "A recursive formation for constrained mechanical system dynamics: Part III, Parallel processing implementation" Mechanisms, Structures, and Machines, 16:249-269, 1988

[12] A. Fijany and A. K. Bejczy, "Techniques for parallel computation of mechanical manipulator dynamics, part II Forward Dynamics" Advances in Robotic Systems and Control, Vol. 40, pp. 357-410, Academic Press, March 1991

[13] I. Sharf, and G. M. T. D'Eleuterio, "An iterative approach to multibody simulation dynamics suitable for parallel implementation" Journal of Dynamic Systems, Measurement and Control, 115:730-735, Dec. 1993

[14] A. Fijany and A. K. Bejczy, “Parallel computation of forward dynamics of manipulators” NASA Technical Brief, Report NPO-18706 12, NASA Jet Propulsion Laboratory, Item # 82 1993

[15] K. S. Anderson, “Efficient modeling of constrained multibody systems for application with parallel computation” Zeitschrift für Angewandte Mathematik und Mechanik, Vol. 73, No. 6, pp. 935-939, 1993

[16] A. Fijany, I. Sharf, and G. M. T. D’Eleuterio. Parallel $O(log\, n)$ algorithms for computation of manipulator forward dynamics. IEEE Transactions on Robotics and Automation, 11(3):389-400, 1995

[17] K. S. Anderson and S. Duan, “A highly parallelizable low-order algorithm for dynamics of multi-rigid-body systems: Part I, chain systems” Mathematical and Computer Modeling, 30: 193-215, 1999

[18] R. Featherstone, “A divide-and-conquer articulated body algorithm for parallel $O(\log_2 n)$ calculation of rigid body dynamics, Part 1: Basic algorithm” International Journal of Robotics Research, 18(9):867-875, Sep. 1999

[19] J. H. Critchley and K. S. Anderson, “A Parallel Logarithmic Order Algorithm for General Multibody System Dynamics”, Accepted for publication in Multibody System Dynamics Journal, 2004

[20] Almassi, G. S. and Gotlieb, A., Highly Parallel Computing, Benjamin-Cummings, Menlo Park, CA, 2nd edition, 1994.

[21] J. H. Argyris and D. W. Scharpf, “Finite elements in time and space” Journal of Royal Aeronautical Society, 73: 1041-1044, 1969

[22] I. Fried, “Finite element analysis of time-dependent phenomena” American Institute of Aeronautics and Astronautics Journal, 7: 1170-1173, 1969

[23] D. H. Hodges and R. R. Bless, “Weak Hamiltonian finite element method for optimal control problems” J. of Guidance, Control and Dyn. 14: 148-156, 1992

[24] M. Borri and C. Bottasso, “Petrov-Galerkin Finite Elements in Time for Rigid-Body Dynamics” J. of Guidance, Control and Dyn. 17(5): 1061-1067, 1994

[25] O. P. Agrawal and V. R. Sonti, “Modelling of stochastic dynamic systems using Hamilton’s law of varying action” J. of Sound and Vibration, 192: 399-412, 1996

[26] A. R. Atilgan, D. H. Hodges, M. A. Ozbek and W. Zhou, “Space-time mixed finite elements for rods” J. of Sound and Vibration, 192(3): 731-739, 1996

[27] S. Lee and Y. Kim, “Time domain finite element method for inverse problem of aircraft maneuvers” Journal of Guidance, Control, and Dynamics 20: 97-103, 1997

[28] D. L. Kunz, “Multibody System Analysis Based on Hamilton’s Weak Principle” 42nd Structures, Structural Dynamics and Materials Conference and Exhibit, AIAA 2001-1296, Seattle, WA, 16-19 April 2001

[29] J. Suk and Y. Kim, “Modelling of vibrating systems using time-domain finite element method” J. of Sound and Vibration, 254(3): 503-521, 2002

[30] T. J. R. Hughes, "The finite element method, Linear static and dynamic finite element analysis" Dover Publications Inc., New York, 1987

[31] P. C. Hughes, "Spacecraft attitude dynamics" Wiley, 1986

[32] K. S. Anderson and Y. Hsu, "Domain Approximation and Deterministic Progression in Genetic Crossover" Engineering Optimization, 33: 683-706, 2001

[33] L.N. Myrabo, "World Record Flights of Beam-Riding Rocket Lightcraft: Demonstration of 'Disruptive' Propulsion Technology" AIAA Paper N. 2001-3798, 37th AIAA/ASME/SAE/ASEE Joint Propulsion Conference, Salt Lake City, Utah, USA, July 2001